DTIC FILE COPY

SECURITY CLASSIFICATION OF THIS PAGE

AD-A217 457

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS: NONE
	3. DISTRIBUTION/AVAILABILITY OF REPORT: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.
	5. MONITORING ORGANIZATION REPORT NUMBER(S): AFIT/CI/CIA-88-244
6a. NAME OF PERFORMING ORGANIZATION: AFIT STUDENT AT University of Illinois — 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION: AFIT/CIA
6c. ADDRESS (City, State, and ZIP Code)	7b. ADDRESS (City, State, and ZIP Code): Wright-Patterson AFB OH 45433-6583
8a. NAME OF FUNDING/SPONSORING ORGANIZATION — 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification) (UNCLASSIFIED)
IMPLEMENTATION AND EVALUATION OF A RESOURCE ALLOCATION ALGORITHM TO DETERMINE THE MINIMUM NUMBER OF INSPECTORS

12. PERSONAL AUTHOR(S)
JOHN T. CLATANOFF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
THESIS/~~DISSERTATION~~	FROM ______ TO ______	1988	241

16. SUPPLEMENTARY NOTATION
APPROVED FOR PUBLIC RELEASE IAW AFR 190-1
ERNEST A. HAYGOOD, 1st Lt, USAF
Executive Officer, Civilian Institution Programs

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

DTIC
S ELECTE FEB 01 1990 D
D

90 02 01 043

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: ERNEST A. HAYGOOD, 1st Lt, USAF	22b. TELEPHONE (Include Area Code): (513) 255-2259 — 22c. OFFICE SYMBOL: AFIT/CI

DD Form 1473, JUN 86 — Previous editions are obsolete. — SECURITY CLASSIFICATION OF THIS PAGE

AFIT/CI "OVERPRINT"

IMPLEMENTATION AND EVALUATION OF A RESOURCE ALLOCATION ALGORITHM TO DETERMINE THE MINIMUM NUMBER OF INSPECTORS

BY

JOHN T. CLATANOFF

B.S., University of Wisconsin, 1976

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1988

Urbana, Illinois

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

THE GRADUATE COLLEGE

MAY 1988

WE HEREBY RECOMMEND THAT THE THESIS BY

JOHN T. CLATANOFF

ENTITLED IMPLEMENTATION AND EVALUATION OF A RESOURCE ALLOCATION ALGORITHM TO DETERMINE THE MINIMUM NUMBER OF INSPECTORS

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF MASTER OF SCIENCE

M. Demsky

Director of Thesis Research

A. M. Clausing

Head of Department

Committee on Final Examination†

Chairperson

† Required for doctor's degree but not for master's.

O 517

ACKNOWLEDGMENTS

I would like to extend my gratitude to Professor Mohammed I. Dessouky, Department of Mechanical and Industrial Engineering, University of Illinois, for all of the guidance and direction he provided. His comments and suggestions were of great value to me in the preparation of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

(1) a_{kw}: time available for an inspector k on day w

(2) A: maximum number of days of inspector availability

(3) d_{ip}: duration of class p inspection at node i

(4) d_k: number of days of available time for an inspector k

(5) D_r: demand requirement to be assigned to an inspector(s)

(6) D_s: demand that is satisfied by an inspector assignment

(7) D_{sk}: the amount of demand an inspector k may satisfy

(8) f_p: number of visits required per working period W

(9) m: number of nodes in a network

(10) n: number of inspectors

(11) P: number of frequency classes p

(12) r_p: number of visits remaining to be assigned in period p

(13) s_{ij}: savings incurred at link (i,j)

(14) t_{ij}: travel time between two nodes i and j

(15) t_k: travel time assigned to an inspector k

(16) T_k: the set of all tours for an inspector k

(17) v_{kp}: number of visits covered by inspector k in period p

(18) W: number of working days

IMPLEMENTATION AND EVALUATION OF A
RESOURCE ALLOCATION ALGORITHM TO DETERMINE
THE MINIMUM NUMBER OF INSPECTORS

By John T. Clatanoff, Major, USAF
1988, 233 pages

Master of Science in Industrial Engineering
from the University of Illinois at Urbana-Champaign

ABSTRACT

An essential element of any logistics system involves the allocation and routing of critical resources which provide goods or services on a regular basis. This thesis presents the problem of determining the minimum number of inspectors required to perform cyclic activities at multiple locations over an arbitrary time period. Formulated as an integer programming problem, it is similar to the broad class of problems known as vehicle routing problems.

A heuristic approach is adopted for solving this inspector problem by essentially dividing it into a load assignment problem and a tour construction problem for every node with demand. The result is an algorithm to solve generic resource allocation problems with any number of depots. A computer-based implementation of the algorithm was developed to evaluate its performance. As with any heuristic approach, there is no guarantee of an optimal solution. However, great care was taken to assure feasible solutions. When inspector utilization factors are considered, the algorithm is shown to be a useful tool for making timely management decisions.

1. INTRODUCTION

1.1 Background

An essential element of any logistics system involves the allocation and routing of critical resources which provide goods or services on a regular basis. This generic situation occurs often in resource allocation applications and is most frequently referred to as the vehicle routing problem (VRP). In the classic VRP, customers with given demand requirements are serviced by a fleet of vehicles stationed at one or more depots where the purpose is to minimize some cost objective. Numerous practical applications have followed, permitting customer demand and vehicles to take on a variety of forms.

In October 1987, the Department of Mechanical and Industrial Engineering at the University of Illinois was contracted to solve a form of the VRP applied to quality assurance inspectors. The problem was to determine the number of inspectors required to perform all inspection activities at twenty-six locations. The inspectors, based at two of the locations, travelled by car to their various inspection sites. The inspectors were divided by their skill specialty, of which there were several. While similar to the basic VRP, constraints were added which prohibited handling the problem using existing VRP solution methods. Consequently, a new methodology was developed by Dessouky, Palekar, and Zaki [12] to determine the number of inspectors needed in each skill category.

Their methodology will be explored in the course of this

study. The algorithm has now been formalized into a completely computer-based implementation. The implementation was then tested to evaluate its effectiveness and utility as a decision making tool for the management of critical resources.

1.2 Objectives

This thesis will present the problem of determining the number of inspectors required to perform cyclic activities at multiple sites over an arbitrary time period. The objectives of this thesis are to:

(a) Extend the single-depot inspector resource allocation model given by Dessouky et al. into a multiple-depot model and modify their methodology to produce an algorithm to solve generic resource allocation problems with any number of depots.

(b) Survey the broad class of problems classified as VRPs, examine some of their heuristic approaches, and show why a new solution method was needed to determine the number of inspectors required for the specific problem to be presented.

(c) Develop a completely computer-based implementation of the modified algorithm to determine the number of inspectors.

(d) Evaluate the computerized algorithm by comparing it performance to results obtained by Dessouky et al. and the lower bounds on the optimum solutions.

(e) Evaluate, through experimentation, the impact of increasing the number of depots and the demand levels on the required number of inspectors and the performance of the algorithm.

Throughout this thesis, inspector resources will be addressed as the critical resource in question. It should be noted that many resource allocation problems could fit the generic model to be presented.

1.3 Problem Statement

Given any set of inspection sites, a map of the locations may be represented in the form of a network. The nodes of the network are used to represent the locations and the arcs depict the travel times between locations. If no arc is present between nodes, then there is no direct route between them.

For any overall time period chosen, inspections are required at the various nodes of the network over a range of frequency periods with each node having different requirements. These requirements can be totaled into total time required per period for each node and tabulated. What is required is the total time spent inspecting at each node under each frequency class for the entire time period. From this, the time per visit per frequency period is easily calculated. It will be assumed that the units for frequency classes are in days per total time span. For example, a weekly inspection over a monthly period would be four days per month. Also, the highest frequency of inspections allowed for a single inspector is once per day.

The time span used to determine total time spent inspecting controls the number of days required for inspecting tasks. This number may be reduced if certain days per period are not days on which inspections are normally performed. For example, if the

time frame is a month, then the number of days dictated by this time span might be 30 days. If however, weekends and holidays are not inspection days, the actual inspection days might be reduced to 21 days. The actual number of inspection days for the desired time span is thus the maximum frequency of daily visits possible for the time covered.

Two constraints affect an inspector's availability for inspection tasks. First, programmed non-availability such as vacations make an inspector's actual availability less than the total days required for the period. Continuing with the monthly example, an inspector with 3 programmed days of non-availability per month has only 18 days left for performing inspector tasks. Secondly, the time available per day for travel and inspection must be established and is expected to be the same for all inspectors in a skill category.

For each skill category, the problem becomes one of determining the travel times, periodic demands, required days for inspection, and inspector availability constraints. The problems are then treated separately. This is permissable as long as there is no interaction between skill categories. When skill categories are combined, the consolidation which results is treated as a separate higher-level skill category to avoid this complication.

1.4 Problem Formulation

The following representation is an integer programming formulation for the problem of finding the minimum number of

inspectors as presented by Dessouky et al. [12] which is extended here to cover any number of depots. The modification of this formulation to n-depots introduces notation similar to that used by Golden, Magnanti, and Nguyen [17].

Let

i or j:	a depot,	$i,j = 1,\dots,D$
i or j:	a node,	$i,j = 1,\dots,D,D+1,\dots m$
k:	an inspector,	$k = 1,\dots,n$
w:	a working day,	$w = 1,\dots,W$
p:	a frequency class,	$p = 1,\dots,P$.

Then, D is the number of depots, m is the number of nodes, n the number of inspectors, W is the number of working days covered, and P is the number of inspection frequency classes. Also, let f_p denote the number of visits per number of working days W required to perform class p inspections.

Given

d_{ip}: duration of class p inspection at node i per visit

h_{ip}: the need for class p inspection at node i

$$h_{ip} = \begin{cases} 1 & \text{if } d_{ip} > 0 \\ 0 & \text{otherwise} \end{cases}$$

a_{kw}: time available for inspector k on day w

t_{ij}: travel time between nodes i and j

Find

y_{ikw}: the incidence of a visit by inspector k to node i on day w

$$y_{ikw} = \begin{cases} 1 & \text{if inspector k visits node i on day w} \\ 0 & \text{otherwise} \end{cases}$$

x_{ijkw}: the sequence in which nodes are visited

$$x_{ijkw} = \begin{cases} 1 & \text{if inspector k visits node j immediately after node i on day w} \\ 0 & \text{otherwise} \end{cases}$$

z_{kw}: the assignment of a visit to inspector k on day w

$$z_{kw} = \begin{cases} 1 & \text{if } \sum_{i=1}^{m} y_{ikw} \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

l_{ikw}: the inspection load of inspector k at node i on day w

b_{iw}: the need for an inspection visit to node i on day w

$$b_{iw} = \begin{cases} 1 & \text{if } \sum_{k=1}^{n} l_{ikw} > 0 \\ 0 & \text{otherwise} \end{cases}$$

s_{ikpw}: the time spent at node i by inspector k to perform class p tasks on day w

The inspector allocation problem is:

Minimize

$$n$$

Subject to

$$\sum_{w=1}^{W} \sum_{k=1}^{m} s_{ikpw} = d_{ip} f_p \quad i=1,\ldots m;\ p=1,\ldots,P \tag{1}$$

$$\sum_{p=1}^{P} s_{ikpw} - l_{ikw} = 0 \quad i=1,\ldots,m;\ k=1,\ldots,n;\ w=1,\ldots,W \tag{2}$$

$$\sum_{w=1}^{W} \sum_{k=1}^{m} y_{ikw} \geq f_p h_{ip} \qquad \text{for all } i \text{ and } p \tag{3}$$

$$s_{ikpw} \leq y_{ikw} M \qquad \text{for all } i,k,p,w \tag{4a}$$

$$s_{ikpw} \leq b_{iw} M \qquad \text{for all } i,k,p,w \tag{4b}$$

$$y_{ikw} \leq z_{kw} \qquad \text{for all } i,k,w \tag{4c}$$

where M is an arbitrarily large number.

$$\sum_{w=1}^{W} z_{kw} \leq A \qquad \text{for all } k \tag{5}$$

where A is the maximum number of days available per inspector.

$$\sum_{k=1}^{n} y_{ikw} \geq b_{iw} \quad i=D+1,\ldots,m;\ w=1,\ldots,W \tag{6a}$$

$$\sum_{k=1}^{n} y_{ikw} \geq n \quad w=1,\ldots,W \tag{6b}$$

$$\sum_{j=1}^{m} l_{ikw} + \sum_{i=1}^{m} \sum_{j=1}^{m} t_{ij} x_{ijkw} \leq a_{kw} \quad k=1,\ldots,m;\ w=1,\ldots,W \tag{7}$$

$$\sum_{j=1}^{m} x_{ijkw} = \sum_{j=1}^{m} x_{jikw} = y_{ikw} \quad i=1,\ldots,m;\ k=1,\ldots,n;\ w=1,\ldots,W \tag{8}$$

$$\sum_{i,j \varepsilon S} x_{ijkw} \leq \sum_{i \varepsilon S} b_{iw} \quad \text{for all } S;\ k=1,\ldots,n;\ w=1,\ldots,W \tag{9}$$

where $S = \{(i,j): \sum_{i \varepsilon Q} \sum_{j \varepsilon Q} x_{ijkw} \leq |Q| - 1\}$

where Q is any non empty subset of $\{D+1,\ldots,m\}$.

$$y_{ikw} \ \varepsilon\ \{0,1\} \qquad \text{for all } i,k,w \tag{10a}$$

$$x_{ikw} \ \varepsilon\ \{0,1\} \qquad \text{for all } i,k,w \tag{10b}$$

$$b_{iw} \ \varepsilon\ \{0,1\} \qquad \text{for all } i \text{ and } w \tag{10c}$$

$$z_{kw} \ \varepsilon\ \{0,1\} \qquad \text{for all } k \text{ and } w \tag{10d}$$

$w_{ikpw} \geq 0$ for all i,k,p,w (11a)

$l_{ikpw} \geq 0$ for all i,k,p,w (11b)

The objective of the above formulation is to minimize the number of inspectors n subject to essentially 11 constraints. The first five constraints restrict the assignment of daily inspection loads. Constraint (1) forces the sum of times spent by inspectors at each site for frequency class p over all days w to equal the total required inspection time at the site for all frequency classes. Constraint (2) defines the time spent at site i by any inspector on a given day w to perform inspections over all frequency periods be equal to the inspection load of the inspector for day w at site i. For each site, constraint (3) requires that the total number of visits to the site at least equal the number of visits required by the highest frequency class with positive demand. Constraints (4a) and (4b) force an inspector to visit a site if inspecting it. Finally, constraint (5) ensures that an inspector does not work more days than the maximum possible.

Constraints (6a) through (9) provide routing restrictions. Constraints (6a) and (6b) make sure that each demand site is served the sufficient number of times. Total elapsed route time is limited by constraint (7) to the time available of an inspector on a given day. Route continuity is represented by constraint (8) insuring that an inspector arriving at a demand site also exits. Lastly, constraint (9) eliminates the possibility of sub-tours, thus guaranteeing a connected path to

the originating location.

Constraints (10) and (11) set the numeric constraints. Constraints (10a) through (10d) restrict the decision variables to integer values. Constraints (11a) and (11b) restrict the time variables to non-negative values.

The multiple depot formulation above represents the mathematical model which has been implemented. Before proceeding to the methodology and implementation which was developed to handle this problem, a survey of the broad class of problems classified as VRPs is necessary to gain insight into the accomplishments and shortfalls of current research.

2. SURVEY OF THE VEHICLE ROUTING PROBLEM

2.1 Introduction

The vehicle routing problem encompasses a broad range of resource allocation problems. Applications have included such diverse areas as school bus routing by Bennett and Gazis [4]; municipal waste collection by Beltrami and Bodin [3]; fuel oil delivery by Garvin, Crandall, John, and Spellman [15], preventive maintenance inspections by Stern and Dror [26]; and railway routing and scheduling by Assad [1]. There is, however, a common denominator among all these various applications. Regardless of the use, the problem is basically one of routing some capacitated resource through a network of demand points to provide goods or services.

According to Magnanti [21], the combinatorial nature of the VRP suggests that it can be formulated and solved, at least in principle, as an integer programming problem. Due to the size and complexity of the problems, however, few studies document the use of integer programming methods for solution. Instead, these problems are typically attacked with heuristic methods.

This chapter will give an overview of the VRP and some of the general heuristic approaches taken. Starting with the well-known traveling salesman, we will build and expand the scope of the problem through the latest methods published for solving a VRP over time. By providing a basis for comparison, it will be seen that while similar, none of the existing methods can be easily modified to solve the problem formulated in Chapter 1.

Finally, a study of various heuristic approaches to the VRP will give insight into the methodology developed by Dessouky et al. and implemented in this thesis.

It should be mentioned at the outset that all of the formulations which follow have objective functions to minimize distance or travel time. The objective function in the inspector resource allocation problem is to minimize the number of inspectors. This difference presents no significant divergence since minimizing the number of inspectors requires minimizing travel time. Hence, we are most interested in an analysis and comparison of the constraints in these formulations.

2.2 The Traveling Salesman Problem

The traveling salesman problem (TSP) is probably the most widely studied version of the VRP. Most vehicle routing models are extensions of the TSP since the routing of a single vehicle through a set of demand points is the most basic component. Thus, the TSP is an excellent place to begin our survey.

The TSP is an extension of the well solved assignment problem. An integer programming formulation of the TSP was first published by Miller, Tucker, and Zemlin [22]. With the added constraint of no sub-tours, the problem becomes highly combinatorial in nature. Applying a formulation provided by Garfinkel [14], we have the following notation which is consistent with the formulation of Chapter 1.

Let

i or j: nodes in the network $i=1,\ldots,m$

x_{ij}: the incidence of travel on arc (i,j) between node i and node j in a tour

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is in a tour} \\ 0 & \text{otherwise} \end{cases}$$

c_{ij}: the cost or distance of traversing arc (i,j)

Minimize

$$\sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} x_{ij}$$

Subject to

$$\sum_{i=1}^{m} x_{ij} = 1 \quad j=1,\ldots,m \tag{12}$$

$$\sum_{j=1}^{m} x_{ij} = 1 \quad i=1,\ldots,m \tag{13}$$

Constraints (12) and (13) along with the objective function define the well-solved assignment problem. The no subtours constraint is enforced mathematically by the addition of the following constraint:

$$\sum_{i \varepsilon S} \sum_{j \varepsilon S} x_{ij} \leq |S| - 1$$

where S is any nonempty, proper subset of all nodes, {1,...,m}. This notation forms the basis for the no subtours constraint of the inspector problem. The problem here is to find a tour through the m nodes which minimizes the total distance or cost where each node is visited exactly once.

2.3 The Multiple Traveling Salesman Problem

The multiple traveling salesman problem (m-TSP) is a

generalization of the TSP and is more applicable to real world models. Given n salesmen and m nodes, the objective is to find n subtours beginning from the origin such that all nodes but the origin are visited exactly once by exactly one salesman and the distance or cost is minimal over all m salesmen. To obtain a m-TSP formulation, we expand the formulation of the TSP given above.

Let

n: the total number of salesmen

$$a_i = \begin{cases} n \text{ if } i = 1 \\ 1 \text{ if } i \neq 1 \end{cases}$$

$$b_j = \begin{cases} n \text{ if } j = 1 \\ 1 \text{ if } j \neq 1 \end{cases}$$

Change

(a) Constraint (12) to

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad j=1,\ldots,m$$

(b) Constraint (13) to

$$\sum_{j=1}^{m} x_{ij} = a_i \qquad i=1,\ldots,m$$

The m-TSP, as shown by Bellmore and Hong [2], can be transformed to the standard TSP and is thus no more difficult than the one-salesman counterpart. The equivalence is obtained by creating n copies of the origin, each connected to the other nodes by the same distances as the original origin. By treating each copy of the origin as unconnected nodes, an optimal single-

salesman tour in the expanded network will never use an arc which traverses two copies of the origin. Thus, when the copies are combined into a single origin, the single tour satisfying the TSP results in the n subtours required by the m-TSP formulation.

2.4 The Basic Vehicle Routing Problem

The vehicle routing problem extends the m-TSP to a capacitated fleet of vehicles (salesmen). The VRP was first considered by Dantzig and Ramser [11] who developed a heuristic approach using linear programming concepts and aggregation of nodes. The problem involves the routing of vehicles with given capacity through a network so that the demand requirements at every node are met and the total travel cost or distance is minimized. The basic VRP schedules one route per vehicle which starts and ends at the depot.

The basic VRP has been applied to a large number of situations by providing the following extensions as summarized by Christofides [9]:

(a) Allow each vehicle to operate more than one tour provided the total time spent is less than some operating time limit.

(b) Allow each node to be visited only during specified "time windows" during a given operating period.

(c) Allow vehicles to operate only during specified time windows.

(d) Allow vehicles to provide both deliveries and collections while servicing nodes. In some cases, it may be

possible to mix deliveries and collections in the same tour. Other situations require "backhauling" where all deliveries in the tour must be made before collections can be performed.

(e) Allow for time consuming activities other than travel such as unloading and/or loading at the nodes, depot loading and/or unloading prior to tour initiation, and queuing times at the depot if loading bays are limited.

While the above extensions illustrate only a subset of the possibilities, they do not change the underlying characteristics of the basic VRP. In addition, the objectives of the problem may be expanded as well. According to Christofides, Mingozzi, and Toth [6], some of these objectives are to:

(a) Minimize the total cost of vehicles used assuming all customer demands must be satisfied.

(b) Minimize the total route distance or time travelled assuming all customer demands must be satisfied.

(c) Minimize the total number of vehicles required assuming all customer demand must be satisfied and using this number of vehicles, minimize their distance or time travelled.

Numerous exact formulations of the basic VRP exist in the literature. A set partitioning formulation set forth by Christofides and Korman [5] allows for the introduction of any number of extensions listed above. The problem has also been modeled as a dynamic programming formulation by Christofides, Mingozzi, and Toth [7]. Integer programming formulations appear to be most prevalent, however. The formulation which follows

attempts to parallel the notation used in the inspector problem and is similar to the one presented by Fisher and Jaikumar [13]. Let

i or j: a node, $i,j=1,\ldots,m$

k: a vehicle, $k=1,\ldots,n$

c_{ij}: the distance or cost of traversing arc (i,j)

q_i: the demand at node i

Q_k: the capacity of vehicle k

x_{ijk}: the sequence in which nodes are visited

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ visits node } j \text{ immediately after node } i \\ 0 & \text{otherwise} \end{cases}$$

y_{ik}: the incidence of a visit by vehicle k to node i

$$y_{ik} = \begin{cases} 1 & \text{if vehicle } k \text{ visits node } i \\ 0 & \text{otherwise} \end{cases}$$

$$b_i = \begin{cases} n & \text{if } i = 1 \\ 1 & \text{if } i \neq 1 \end{cases}$$

Assume that all vehicle capacities and customer demands are ordered in descending order of q_i and Q_k respectively. The basic single-depot VRP is:

Minimize

$$\sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} \sum_{k=1}^{n} x_{ijk}$$

Subject to

$$\sum_{k=1}^{n} y_{ik} = b_i \qquad i=1,\ldots,m \qquad (14)$$

$$\sum_{i=1}^{m} q_i y_{ij} \leq Q_k \qquad k=1,\ldots,n \qquad (15)$$

$$\sum_{j=1}^{m} x_{ijk} = \sum_{j=1}^{m} x_{jik} = y_{ik} \qquad i=1,\ldots,m;\ k=1,\ldots,n \qquad (16)$$

$$\sum_{i,j \varepsilon S} x_{ijk} \leq |S| - 1 \qquad k=1,\ldots,n \qquad (17)$$

where S is any nonempty proper subset of $\{2,\ldots,m\}$.

$$y_{ik} \ \varepsilon\ \{0,1\} \qquad \text{for all i and k} \qquad (18a)$$

$$x_{ijk} \ \varepsilon\ \{0,1\} \qquad \text{for all i,j,k} \qquad (18b)$$

Constraint (14) ensures that every node other than the depot is allocated to some vehicle. Constraint (15) is the capacity constraint for each vehicle. Constraint (16) ensures that a vehicle which visits a node also departs the node. Lastly, the subtour elimination stipulation is enforced by constraint (17). Integer constraints (18) are essentially the same as in the inspector problem.

The basic VRP model illustrates the foundation for the inspector problem. Route continuity and subtour elimination constraints are exactly the same in the two problems. In the inspector problem, the capacity of the inspector is the daily time available. The basic VRP provides for one capacitated vehicle satisfying all demand for every node visited whereas any number of inspectors may be required to satisfy demand for a node in the inspector problem. The major element missing in this formulation is the element of time as seen by the absence of any day indices (subscript w in the inspector allocation problem presented in Chapter 1). Thus, existing solution methods for the

basic VRP would fall far short in attempting to solve the inspector problem.

2.5 The Period Routing Problem

The basic VRP is limited to minimizing the cost of distribution for a single day. In the period vehicle routing problem (PVRP), the objective is to develop a set of routes for each day of a given time period where any node of the network may require one or more visits during the period. For a node with f visits required, the visits must occur in some f-day combination over the total time period. The PVRP can be viewed as a sequence of daily subproblems where the intent is to choose a day combination for each node which minimizes total cost. An exact mathematical formulation for the single-depot PVRP is presented here which attempts to adapt the notation developed by Christofides and Beasly [8] to the inspector problem as much as possible.

Let

q_i: the demand for node i per visit

t_{ij}: the travel time between nodes i and j

i or j: a node, $i,j = 1,\ldots,m$

S: any proper subset of $\{2,\ldots,m\}$

w: a day, $w = 1,\ldots,W$

k: a vehicle, $k = 1,\ldots,n$

C_i: the set of allowable day combinations for node i

Q_k: the capacity of vehicle k

D_k: the maximum travel time for vehicle k

R_w: the set of available vehicles on day w

$$h_{ip} = \begin{cases} 1 & \text{if the pth combination is chosen for node i} \\ 0 & \text{otherwise} \end{cases}$$

$$b_{pw} = \begin{cases} 1 & \text{if day w is in day-combination p} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{iw} = \begin{cases} 1 & \text{if node i is visited on day w} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ijkw} = \begin{cases} 1 & \text{if vehicle k travels between nodes i and j on on day w} \\ 0 & \text{otherwise} \end{cases}$$

The PVRP is:

Minimize

$$\sum_{w=1}^{W} \sum_{i=0}^{m} \sum_{j=0}^{m} \sum_{k=1}^{n} t_{ij} x_{ijkw}$$

Subject to

$$\sum_{p \varepsilon C_i} h_{ip} = 1 \quad \text{for all } i > 1 \tag{19}$$

$$\sum_{p \varepsilon C_i} h_{ip} b_{pw} = y_{iw} \quad \text{for all } w, i > 1 \tag{20}$$

$$\sum_{k \varepsilon R_w} x_{ijkw} \leq (y_{iw} + y_{jw})/2 \quad \text{for all } i, j (i \neq j), w \tag{21}$$

$$\sum_{i=1}^{m} x_{itkw} = \sum_{j=1}^{m} x_{tjkw} \quad \text{for all } t, k \varepsilon R_w \tag{22}$$

$$\sum_{k \varepsilon R_w} \sum_{i=1}^{m} x_{ijkw} = \begin{cases} y_{jw} & \text{for all } w, j > 1 \\ |R_w| & \text{for all } w, j = 1 \end{cases} \tag{23}$$

$$\sum_{i \varepsilon S} \sum_{j \varepsilon S} x_{ijkw} \leq |S| - 1 \quad \text{for all } w, k \varepsilon R_w \tag{24}$$

$$\sum_{j=2}^{m} x_{1jkw} \leq 1 \quad \text{for all } w, k \varepsilon R_w \tag{25}$$

$$\sum_{i=2}^{m} q_i \left\{ \sum_{j=1}^{m} x_{ijkw} \right\} \le Q_k \quad \text{for all } w, k \varepsilon R_w \tag{26}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{m} t_{ij} x_{ijkw} \le D_k \quad \text{for all } w, k \varepsilon R_w \tag{27}$$

$$h_{ip} \, \varepsilon \, \{0,1\} \quad \text{for all } i, k \varepsilon S_i \tag{28a}$$

$$x_{ijkw} \, \varepsilon \, \{0,1\} \quad \text{for all } i,j,w,k \varepsilon R_w \tag{28b}$$

Constraint (19) ensures that only one day combination is chosen for each node. Constraint (20) ensures that a node is visited only on a day which includes a visit to the node by the day combination chosen. Constraint (21) prohibits vehicles from traveling between two nodes on a given day unless both require visits on that day. Route continuity is represented by constraint (22) which ensures that a vehicle arriving at a node also exits. Constraint (23) forces a visit to each node on the days it is required to be visited. The standard subtour elimination condition is provided by constraint (24). Constraint (25) restricts the use of any particular vehicle to once. Constraints (26) and (27) set the limits on vehicle capacity and driving time respectively. Finally, constraints (28a) and (28b) are integrality conditions.

The PVRP formulation extends the basic VRP to a problem quite similar to the inspector problem. The set R_w can be viewed as a set with n vehicles over all days w. The route continuity and subtour elimination restrictions for resource k on any day w are the same between the two problems. Further, the vehicle travel time limit, D_k, is equivalent to the time available of an

inspector, a_{kw}, on day w. Treatment of resource capacity, although not equivalent, is also comparable. Vehicles are limited to a maximum of one tour per day with a total capacity Q_k while inspectors are limited to any number of tours per day provided the time spent inspecting and traveling does not exceed a_{kw} and the maximum number of days, A, is not exceeded.

The major difference between the PVRP and the inspector allocation formulations, hence limiting the PVRP, occurs in the handling of the visits over time. While both formulations deal with satisfying demand at nodes over time with capacitated resources, the PVRP assumes the subscript p to be a day combination from the set of possible day combinations at node i, S_i. The demand at node i, q_i, is constant for each visit. The inspector problem makes no such assumption regarding day combinations. Instead, p represents the frequency period for which demand exists at a given node. The demand at node i is a function of the frequency period given by d_{ip}. Any node may require visits ranging from, at most, daily frequency to lower frequency periodic visits. Consequently, we see major differences between the two formulations regarding constraints for the assignment of resources. Thus, changing demands over multiple frequency periods is an extension not easily handled by the PVRP formulation above.

Other more subtle differences are also present. In the PVRP, vehicles assigned to a route satisfy all the demand at each node visited. This constraint is relaxed in the inspector

problem, adding greater complexity. Next, the depot in the PVRP is modeled to have zero demand while the inspector problem provides for satisfying demands over all frequency periods at the depot(s). Finally, the PVRP is intended for shorter time periods than the inspector problem. As the time period is increased, the number of day combinations explodes. The inspector problem was formulated for a problem with annual requirements but can be applied to periods both longer and shorter without degrading performance.

2.6 Heuristic Approaches

Attempting to solve a PVRP or the inspector problem using exact methods would be extremely complex except for the most trivial cases. A tremendous number of variables would be involved (x_{ijkw}, for example). Thus, most of the work done in solving VRPs has been directed at devising heuristics. The purpose of this section is to provide some of the general principles of these heuristic procedures which were considered in the development of a computer implementation for the inspector solution methodology.

According to Christofides, Mingozzi, and Toth [6], the heuristics applied to the basic VRP for the construction of tours can be classified according to: (1) the criterion used to expand tours and (2) whether the tours are constructed sequentially or in parallel.

The criterion for tour expansion is a function defined over all nodes that is used to determine which node should enter a

tour and in what position. The node which optimizes the criterion function is included. Some of the most frequently used criteria are:

(a) The "savings" in distance or cost incurred by including a node in a tour as compared to treating the node separately from the tour. The savings algorithm of Clarke and Wright [10] is the most widely known method using this criterion.

(b) The "extra mileage" of including a node in a tour between two consecutive nodes.

(c) The radial position determined by the angle formed between the ray from the depot to a node and the ray from the depot to a node in a tour. The sweep algorithm of Gillett and Miller [16] uses this criterion.

(d) A composite function including some or all of the above along with added criteria such as the quantity of demand to be satisfied to a node.

Tours are also constructed sequentially or simultaneously. In sequential algorithms, tours are constructed one at a time and expanded according to some criterion until all the nodes are visited. No decision is made regarding whether a node should be included in one tour or another. This decision is made implicitly by the criterion function which orders the nodes for inclusion into the tour currently being constructed.

Simultaneous tour construction can have either a fixed number of tours or an indefinite number of tours. If a fixed number of tours is established prior to the procedure, each node

is included to one of a given number of tours. The number of tours at termination will be this fixed number of tours. When the number of tours is not fixed ahead of time, small tours (initially a round-trip from the depot to a single node) are built into larger ones with additional nodes being entered until the tour can grow no more. The number of tours at termination will be unpredictable in this case.

All of the heuristic approaches developed for the PVRP use the concept of day combinations. Beltrami and Bodin [3] considered two approaches regarding hoist compactor routing. In the first, routes are developed and assigned to days of the week. The second approach randomly assigns nodes to days of the week and then develops routes.

Russell and Igo [25] specify a required number of visits for each node within a time period rather than using explicit day combinations. Three heuristic approaches were considered.

Their first heuristic attempts to generate a feasible solution to the load assignment problem. The procedure starts by assigning nodes requiring the highest number of visits or nodes with specific day requirements. The resulting nuclei of points act to attract other unassigned nodes. Nodes are assigned sequentially in order of their frequency of demand. Once the delivery combinations have been decided, a VRP algorithm can be applied to each day of the period separately.

Their second approach uses the routes developed in the first heuristic and attempts to make improvements through link

exchanges using a modification of the MTOUR algorithm developed by Russell [24]. The MTOUR algorithm, modified for the PVRP, identifies links which satisfy all constraints and minimizes the total length of the routes. These links then replace links from the current feasible solution. Feasible link exchanges occur until further improvements cannot be made.

The last approach modifies the savings algorithm of Clark and Wright [10]. Initially, the savings algorithm has all nodes of a network connected to the depot. For every possible pair of nodes, there is a corresponding savings s_{ij} which is a measure of the savings obtained if nodes i and j are joined on the same tour. If nodes i and j are connected as in Figure 2.1, then the savings $s_{ij} = t_{i1} + t_{1j} - t_{ij}$. Figure 2.2 shows the resulting tour. The savings algorithm modified for the PVRP connects two nodes i and j with the largest s_{ij} if the capacities pertaining to load and distance are not exceeded and delivery spacing constraints are not violated. Routes are then formed using this savings algorithm.

Regardless of the approach taken to obtain feasible tours, once they are developed, tour improvement procedures used for the TSP may be applied. Some of the best known tour improvement heuristics are branch exchange heuristics. These include the 2-opt and 3-opt procedures developed by Lin [19] and the k-opt heuristic ($k \geq 3$) presented by Lin and Kernighan [20]. The branch exchange heuristics, as given by Golden, Bodin, Doyle, and Stewart [18], are applied as follows:

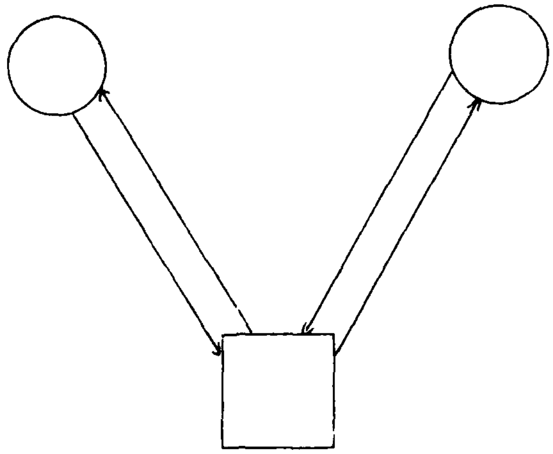


Figure 2.1 Initial Tours

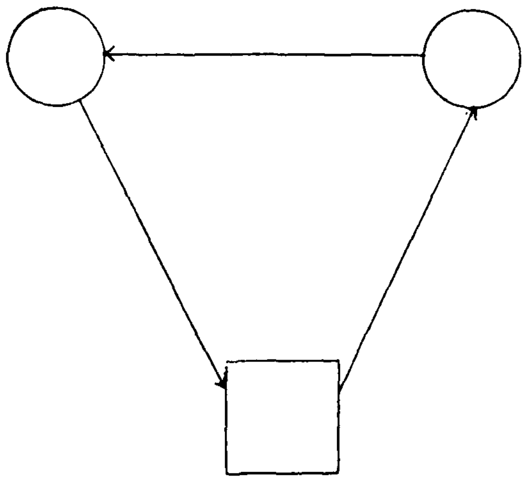


Figure 2.2 Resulting Tour when Savings Possible

(1) Find an initial tour. This tour is often chosen randomly from the set of all possible tours.

(2) Improve the tour using one of the branch exchange heuristics.

(3) Continue (2) until no additional improvement can be made.

The branch exchange procedure terminates at a local optimum. For any given k, the k-change of a tour in (2) consists of deleting k branches in a tour and replacing them with k others to form a new tour. Thus, in (3), a tour is considered k-optimal if it is not possible to improve the tour via a k-change. Generally, the greater the value of k in the k-opt exchange, the stronger the procedure and the better the local optimum is at predicting the true optimum.

These branch exchange procedures are used as a general heuristic approach in many of the VRP and PVRP solution methods. They have also been used successfully to obtain excellent results to large scale TSPs in a reasonable amount of time. However, the approach is computationally expensive, particularly as k is increased. Thus, composite procedures are developed where the idea is to obtain a good initial solution before applying the 2-opt exchange procedure. The 3-opt procedure is then applied to the tour found after the 2-opt exchange. In this way, the number of computationally expensive 3-opt exchanges are reduced. Many variations to the composite procedure are possible but a key factor in the strength of any variation is how good the initial

solution is.

From the discussion in this section, it should be clear that while the inspector allocation problem is a form of the PVRP, the exact nature of its constraints are unlike any seen in the formulations presented in the literature. Thus, we expect existing solution methods for the VRP and PVRP to be insufficient for solving the inspector allocation problem. Fortunately, the heuristic approaches presented here are useful in developing and implementing the methodology to solve the inspector problem. While not used explicitly, they still provide a sound basis for attacking the problem and formalizing the approach.

3. THE SOLUTION ALGORITHM

3.1 Approach

Proposed techniques for solving VRPs have fallen into two groups: those which provide exact solutions using integer optimization methods, particularly branch and bound techniques, and those which provide approximate solutions using heuristics. Due to the large number of complex constraints in the inspector problem formulation of Chapter 1, discrete optimization methods are considered computationally prohibitive. Accordingly, a heuristic approach is adopted for solving the inspector problem.

The approach essentially divides the problem into a load assignment problem and a tour construction problem for every node with demand. Inspectors are added sequentially and for each node requiring inspections, a demand load is assigned to an inspector and a tour determined over the inspector's available working days. The current depot being considered becomes the base for any inspector added. Tours for each inspector are constructed in parallel using a modification of the savings algorithm of Clarke and Wright [10]. This procedure is continued until the demands for an entire depot block (a depot and all nodes assigned to the depot) are satisfied over all frequency periods. The depots and the nodes which form each depot block are assumed to have been identified at the outset. The next depot block is then considered in the same manner until all depot blocks are satisfied.

Inspectors assigned to a depot may be used to satisfy

demands at nodes of other depots. However, by first considering all nodes linked to a depot before proceeding to the next one, inspectors are assigned to satisfy all demand at their local block of nodes before being allowed to travel to the nodes of other depots.

The overall objective of minimizing the number of inspectors in conjunction with the total time travelled requires that the load assignment problems and tour construction problems be considered together. The demand that an inspector may satisfy at a given node is directly related to the time spent traveling. The tour generating approach that is used forms tours as nodes are assigned. The computer implementation of this algorithm seeks no further improvement of inspectors' tours. However, the importance of incorporating link exchange heuristics into the algorithm for tour improvement cannot be overstated.

The solution procedure to be presented will determine the minimum number of inspectors for any number of depots over any specified time period. For this time period, the number of working days requiring inspectors is W, the number of days of inspector availability is A, and their daily time available is a_{kw}. These parameters are given. In addition, demand, d_{ip}, at node i may range over multiple frequency periods j within the specified time period.

Based on these considerations, the following guidelines are incorporated into the computer implementation:

(a) Depots are handled sequentially as they are ordered in

the data. All demand at nodes linked to a given depot are satisfied before considering the next depot.

(b) Locations with the most frequent non-zero demand (smallest p) are considered first. Thus, inspectors are first allocated to nodes with daily demand followed by those with less frequent demand.

(c) Inspectors assigned to visit a location for a periodic inspection are capable of satisfying lower-frequency demands at the same location without additional travel time, provided they have sufficient slack time.

(d) Additional inspectors are assigned when needed to satisfy demands of any frequency. The depot of current consideration becomes the one where the new inspector is based.

3.2 Procedure

The following procedure was developed by Dessouky, Palekar, and Zaki [12] to determine the minimum number of inspectors, given a single depot:

1. Find the minimum number of inspectors required to satisfy demand at each location with daily demands. Construct shortest time tours if needed.

2. Compute the slack time available per day for every inspector and the number of days over which this slack time applies.

3. Select the location which has the most frequent non-daily demand among all locations with unsatisfied demands. Try to meet the non-daily demand at this location using the slack

time computed in (2) for available inspectors. Otherwise, assign a new inspector to meet this demand. Update slack times. Construct shortest time tours if needed.

4. Repeat (3) until demand is satisfied at every location.

5. After each inspector has been added and assigned, combine tours if possible to reduce total travel time.

The proposed algorithm formalizes the procedure above and extends it to the multiple depot case. Since tour information is used to determine the amount of demand satisfied and slack time consumed, a modification of the savings algorithm is used in generating tours for both the daily and non-daily assignment procedure. Also, nodes are grouped by depot blocks with inspectors assigned to satisfy both daily and non-daily demand for the entire block before proceeding. This is in contrast to assigning inspectors for every node with daily demand prior to proceeding to the lower frequency demand groups. The modified algorithm is as follows:

(1) **Determine nodes to be satisfied.** For the current depot, proceed to (2) considering the current depot and all nodes linked to it.

(2) **Make daily assignments.** For nodes with positive daily demand, assign inspectors with sufficient slack time and form tours of maximum savings if needed. If there are no available inspectors, assign new inspectors based at the current depot and form tours of maximum savings if needed. Adjust the amount of slack time per day for the inspectors assigned.

(3) **Make non-daily assignments for nodes visited daily.** For the nodes visited in (2) with positive non-daily demand, find inspectors with availability over the total working period, adding new inspectors based at the current depot if necessary. Find another combination of inspectors with availability over the total working period if slack time becomes insufficient for any inspector. Form maximum savings tours if the inspector is not assigned to visit the node daily during the inspector's period of availability. Adjust the slack time per day for the inspectors assigned.

(4) **Make remaining non-daily assignments.** For the other unassigned nodes with positive non-daily demand, assign inspectors in the same manner as (3) above, working from the highest *non-daily frequency* to the lowest. Form maximum savings tours if the inspector is not already assigned to visit the node in a higher frequency period during the inspector's period of availability. As nodes with non-daily demand for a given frequency period are considered, satisfy all of their lower frequency demands prior to proceeding to the next period.

(5) **Perform tour improvements.** After each inspector is added and assigned, use a suitable link exchange heuristic, such as the 2-opt procedure given by Lin [19], to reduce overall travel time. Any reductions of travel time are translated back to additional inspector slack time.

(6) **Step to the next depot.** Continue (1) through (5) until all depots and their nodes have been considered.

3.3 Algorithm

In this section, the proposed algorithm is described in greater detail. A depiction of the basic flow for this algorithm is shown is Figure 3.1.

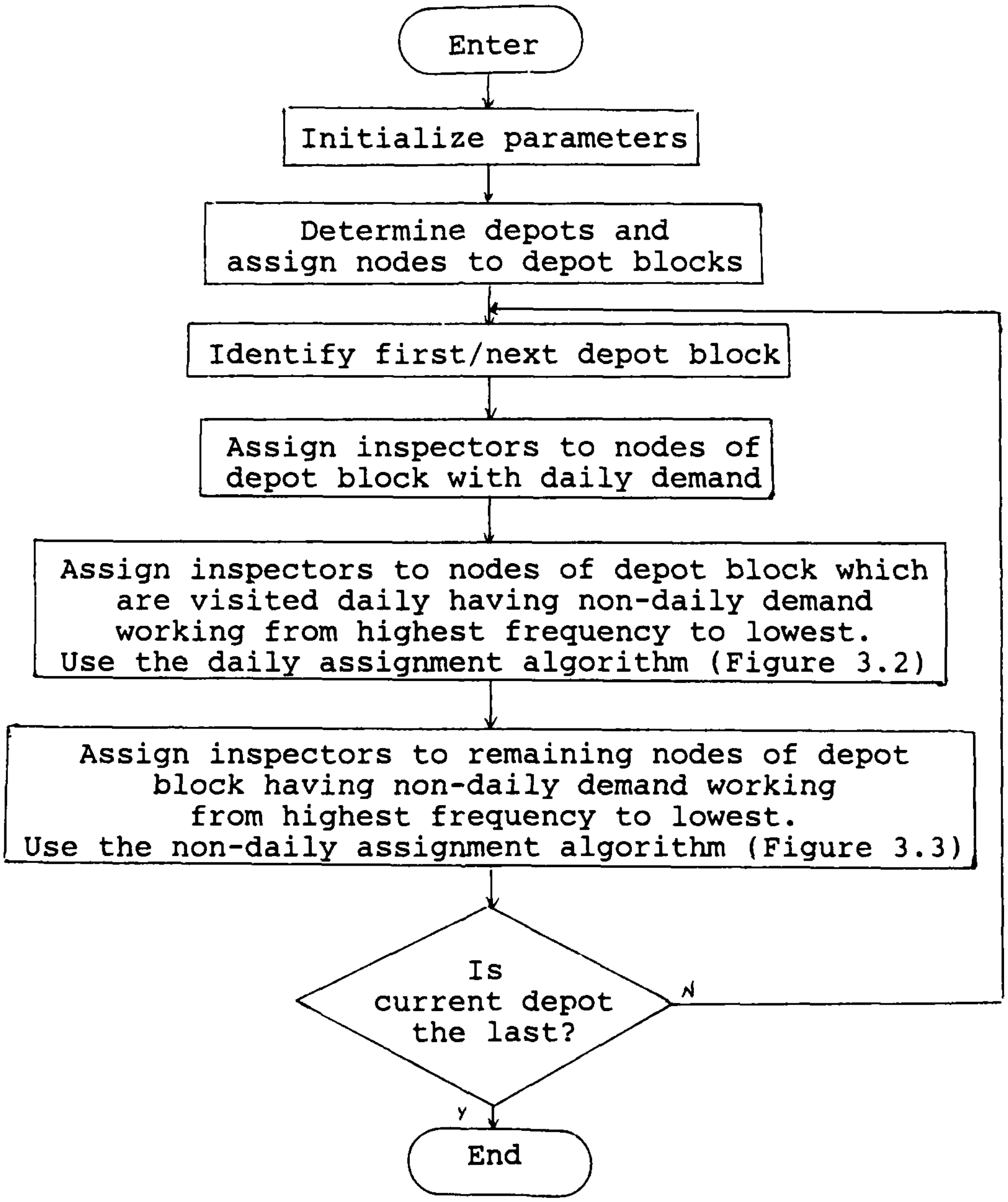


Figure 3.1 Algorithm Flowchart

In the initialization step, all of the given parameters from

the formulation in Chapter 1 must be assigned. These parameters include the following:

(1) m, the number of nodes in the network

(2) D, the number of depots in the network

(3) $B = \{d : d = 1,...,D\}$, the set of depots in the network

(4) W, the number of working days in the time period

(5) P, the number of frequency classes

(6) f_p, the frequency of visits over the total time period required to perform class p inspections

(7) d_{ip}, the duration of a class p inspection at node i

(8) h_{ip}, the need for a class p inspection at node i

(9) a_{kw}, the time available for inspector k on day w

(10) t_{ij}, the shortest path time between nodes i and j

The times t_{ij} used in the implementation of this algorithm reflect shortest path times rather than the incidence of arc (i,j). Thus, even if a direct route does not exist between two nodes, a shortest path time exists for every combination of nodes. Shortest paths can be determined using any shortest distance algorithm to determine both the optimal time and path between any two nodes of a network. Representing t_{ij} in this manner requires that the optimal routings be known at this time.

The multiterminal shortest-route algorithm due to Floyd was selected for obtaining this information. Using the code provided by Phillips and Garcia-Diaz [23], the required information is provided in two matrices. The first gives the shortest path time between any node i and j which we include as t_{ij}. The second

matrix identifies the intermediate nodes (if any) of the shortest chains. The shortest chain between two nodes is then constructed by finding the next node in the chain. Let r_{ij} denote the next node in the chain between two nodes i and j. If r_{ij} is j, the path is determined. Otherwise, reenter the matrix to find the next node in the shortest path between r_{ij} and j until j is reached.

The determination of the nodes to be designated as depots and the assignment of nodes to each depot can be accomplished in a number of different ways. Nodes with the greatest number of links to other nodes or nodes which are "centrally located" are often chosen as depots. Depending on the problem, the depots may already be determined. Similarly, depot blocks may be given or some selection criteria may be needed. Perhaps the most common assignment heuristic is to assign nodes to a depot based on the least travel time. What is important here is that some assignment procedure is needed when the depots or the depot blocks are not given.

Once the depots are known and the assignments made, each depot block is assigned inspectors according to how they are ordered in the set B. Inspectors stationed at depots previously assigned may be used to satisfy demand of other depots provided slack time is sufficient. Assignment of daily demands are performed first.

For every node of a given depot block (including the depot) requiring daily visits, inspectors are assigned according to the

daily demand assignment procedure. The flowchart of this algorithm is shown in Figure 3.2. The approach used is a load assignment procedure based on travel times and amount of demand which can be satisfied as determined by the tour which is generated or assigned. It closely parallels the procedure presented by Dessouky et al [12].

This procedure is called for every node i in a given depot block with positive daily demand. Let D_r denote this total daily demand requirement. The day, w, is initialized to one, the first day of the total working period. Inspectors who have been added previously and have slack time remaining are then considered for daily assignment to node i. If the travel time t_k for some inspector k with slack is greater than a_{kw}, the daily time available of inspector k on day w, the inspector has insufficient slack time to satisfy any of the demand D_r. In this case, other inspectors are considered in the same manner until either an inspector with sufficient slack time is found or a new one is added.

Each time an inspector is added, the inspector is assigned to the depot currently being considered. The new inspector's first day of availability is set to the day after the last working day of the previous inspector added. If no inspectors have been added, the first day is set to one. Also, if the total days of inspector availability are less than the total number of working days, an idle period is established immediately after A days of availability are assigned. In this way, inspectors are

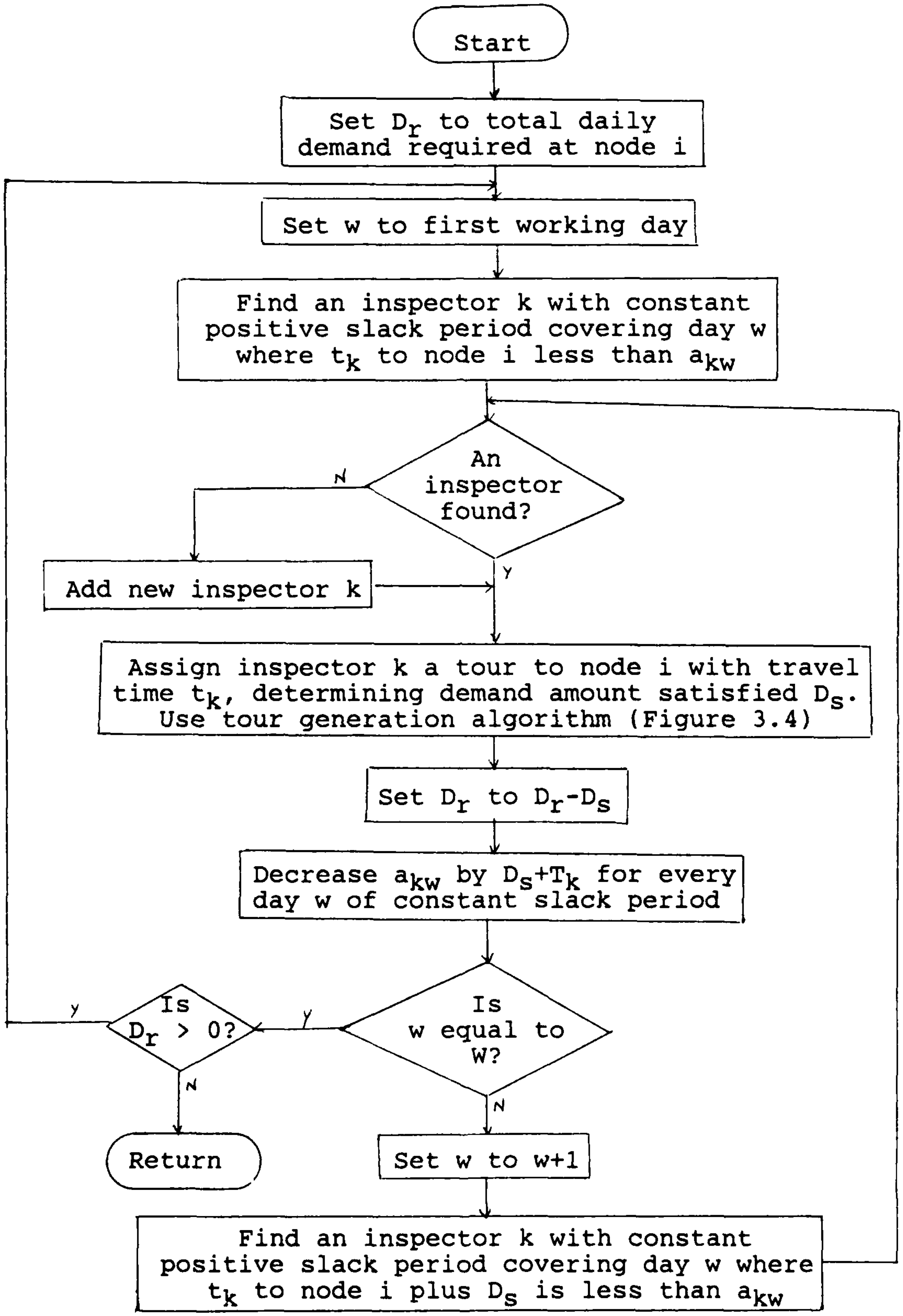


Figure 3.2 Flowchart for Daily Assignments

added in a continuous fashion, eliminating gaps in coverage.

Once some inspector k is obtained with sufficient slack time, the period over which the slack time remains constant is determined and a tour assignment is made. As a tour assignment is made, the amount of daily demand that can be satisfied by the inspector, D_s, is determined. The total daily demand D_r is decreased by D_s to denote the updated demand remaining. The amount D_s is now met at node i for each day of the inspector's constant slack availability period. The daily time available of inspector k is reduced by the amount D_s+T_k by incrementing w by one for each day in the slack availability period.

When w reaches the last day of the inspector's availability period, either more days remain to be assigned or the last working day is reached. When more days remain to be assigned, another inspector with availability immediately after day w is sought whose slack time is sufficient to satisfy D_s demand and travel to node i. The added constraint that the inspector have slack to meet the amount of demand previously satisfied ensures that D_s is satisfied uniformly over all working days and the amount D_r continues to accurately reflect the total daily demand remaining. If none of the previously added inspectors have the time available to meet this requirement, another inspector is added. When w has reached the last working day and D_r is greater than zero, the procedure is repeated starting with the first day. Eventually, D_r is reduced to zero and all daily demand is satisfied using this procedure.

The main algorithm next assigns inspectors to nodes with non-daily demand requirements, starting with the highest frequency class and working to the lowest. In this way, inspectors with sufficient slack time who visit nodes in a higher frequency period meet lower frequency demand without incurring additional travel time. Thus, the procedure affords the maximum opportunity for efficient utilization of inspector slack time. This philosophy also explains why nodes of a given depot block which are visited daily are satisfied before other nodes with non-daily demand.

The assignment of inspectors to meet non-daily demand requirements is the same for nodes of a depot block which are visited daily and those which are not. This procedure, which is depicted in Figure 3.3, modifies the load assignment method used by Dessouky et al. to accommodate multiple depots. In their method, when periodic assignments are made for a given inspector, the equivalent number of slack days are calculated and removed from the inspector's slack availability period. In this procedure, when the same assignments are made, the amount of slack time used per day is determined and subtracted from the inspector's daily time available over the entire availability period.

The non-daily assignment procedure for any node i of a depot sets D_r to the amount of time required per class p visit. The first requirement for making any a non-daily assignment using the slack time of inspectors is to find a set of inspectors with

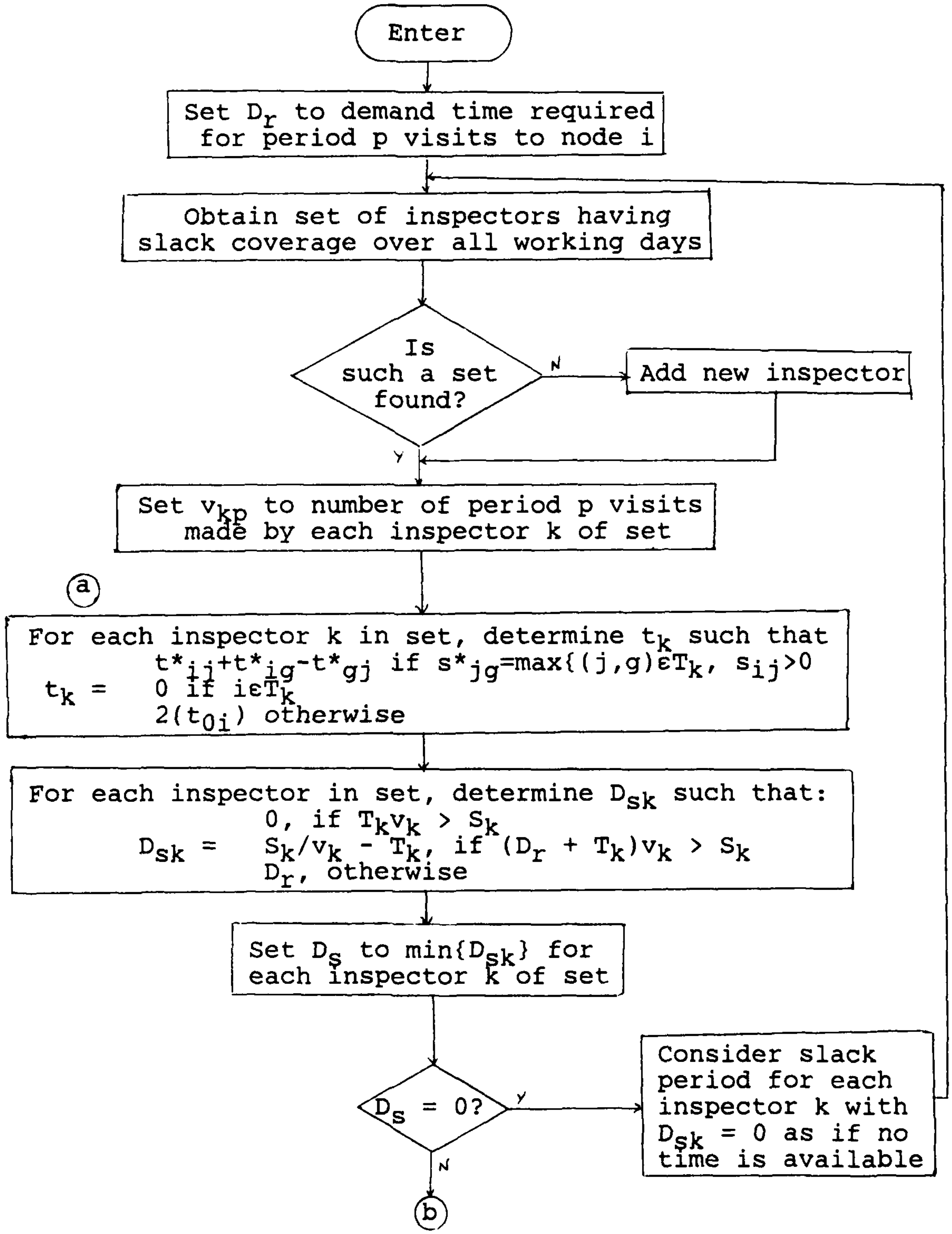


Figure 3.3 Flowchart of Non-daily Assignments

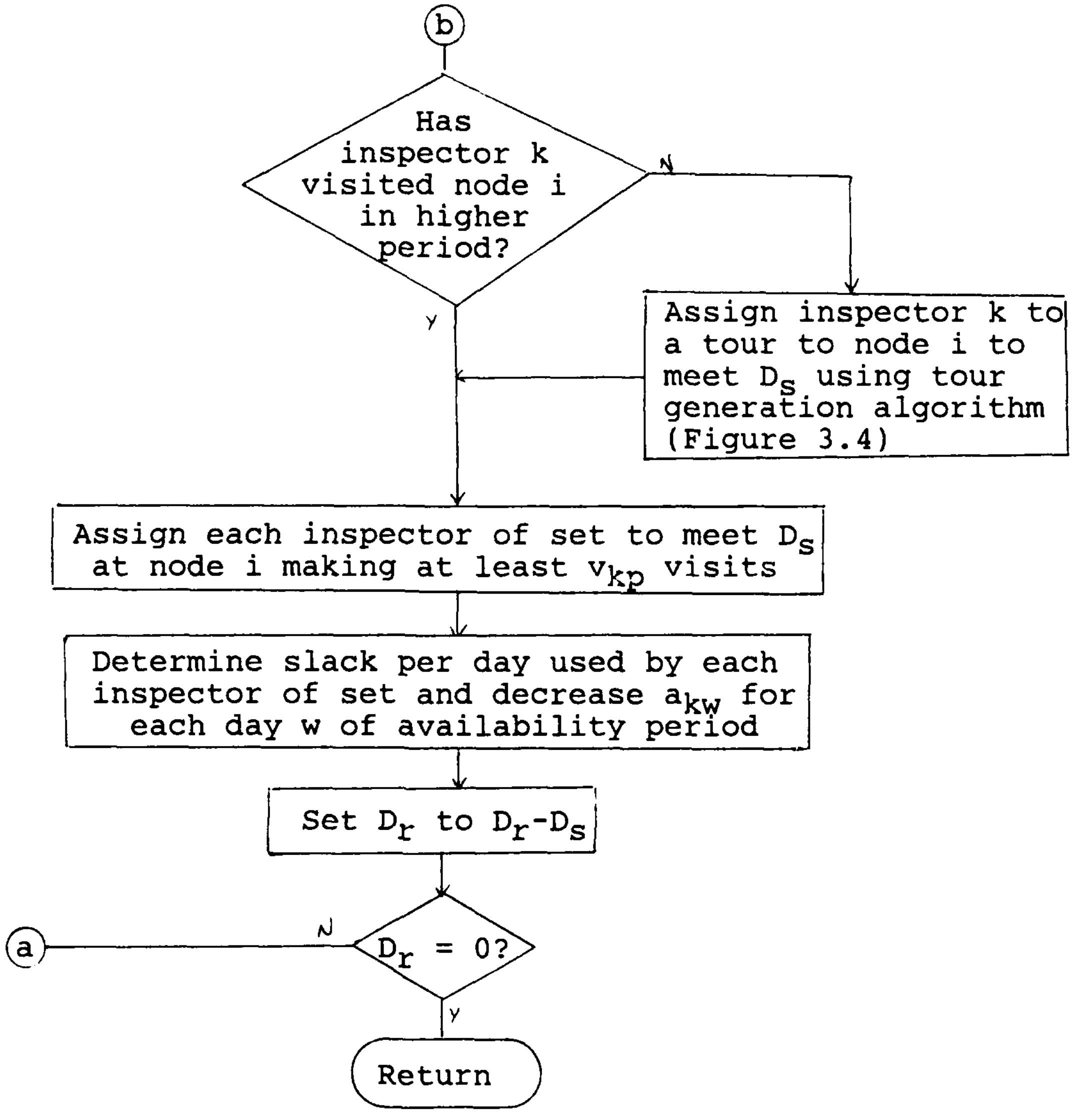


Figure 3.3 (Continued)

positive slack times covering all working days. This set will be one or more inspectors such that their constant slack periods are adjacent and, when combined, include at least every working day. If complete coverage cannot be achieved by existing inspectors, additional inspectors are added until complete coverage is attained. The addition of new inspectors follows the same

approach as used in the daily assignment procedure.

Once we have a set of inspectors with complete coverage, the numbers of period p inspections that can be accomplished by each inspector are calculated. Let d_k denote the number of days of available time of inspector k throughout a given slack period, then d_k may be expressed as

$$d_k = \sum_{w=1}^{W} u_{kw}$$

such that

$$u_{kw} = \begin{cases} 1 \text{ if } a_{kw}>0 \\ 0 \text{ otherwise.} \end{cases}$$

This method guarantees that the visits occur at different times because of the requirement that the joint slack availability periods of all inspectors in the set be sufficient to cover demand for all frequencies. Now recall that f_p denotes the number of visits required over all working days for class p inspections and W represents the total number of working days. To ensure that exactly f_p visits are assigned during frequency period p, let r_p denote the number of visits remaining to be assigned in the period. Initially, r_p is set equal to f_p. Then the number of visits that can be performed by inspector k in frequency period p, v_{kp}, is given by

$$v_{kp} = [\min \{ d_k f_p/W, r_p \}]$$

such that [z] denotes the greatest integer less than or equal to z, and r_p is updated by

$$r_p = r_p - v_{kp}$$

for each inspector k.

For an inspector of the set who has already visited node i in a higher frequency period, no additional travel time is required and t_k is set to zero. Otherwise, it is set to the travel time required for inspector k to visit node i.

The amount of demand that can be met by each inspector k of the set, D_{sk}, is then determined. Let S_k be the total slack time available for inspector k in the slack period where

$$S_k = a_{kw}d_k.$$

Then if the total amount of slack time required to travel to node i by each inspector k to meet period p demand, t_kv_{kp}, is greater than the total slack time available, S_k, the amount of demand that can be satisfied, D_{sk}, is zero. If there is sufficient slack time to travel, the total slack time required, $(D_r+t_k)v_{kp}$, is compared to total slack available, S_k. If total slack required is greater than slack available, the amount of demand that can be satisfied is reduced to

$$D_{sk} = S_k/v_{kp} - t_k$$

which consumes all remaining slack time of inspector k. Otherwise, the inspector has sufficient slack time to meet all demand at node i and D_{sk} is set to D_r.

Unlike the daily assignment procedure where D_s is determined at the time of tour assignment, the actual amount of demand to be satisfied for non-daily assignments is assigned in the non-daily assignment routine. D_s is defined as the minimum amount of demand that can be satisfied by any inspector of the set. If an inspector can satisfy no demand, their slack period does not

permit travel and is removed from further consideration even though some positive slack may exist. In this case, D_s is zero and the procedure reverts to the beginning to find another set of inspectors with total coverage.

Assuming D_s is greater than zero, this amount of demand is assigned to each inspector of the set over their respective availability periods, performing their share of the total number of visits. If node i has already been visited to meet higher frequency demand, a new tour is not needed. Otherwise, a tour is assigned to meet demand of D_s.

The daily availability of each inspector of the set is reduced by the equivalent amount of slack time used per day over their respective slack periods. Recall that d_k denotes the number of days in the constant slack period of inspector k, D_s is the amount of demand satisfied by each inspector of the set, t_k is the travel time required of inspector k to node i, and v_{kp} is the number of period p visits to be performed by inspector k. Then, the slack per day for any inspector k of the set is

$$\text{slack per day} = (D_s + t_k)v_{kp}/d_k.$$

In this way, assignment the demand satisfied is accomplished uniformly for each working day. D_r is then reduced by the amount of demand satisfied. If D_r is reduced to zero, all period p demand is met and the procedure terminates. Otherwise, it is repeated until all remaining demand is assigned.

Both the daily and non-daily assignment routines apply the same tour generation approach. When a specific inspector is

assigned, the tour to which the inspector is assigned must cover the precise number of visits required of the inspector for the frequency period and be applicable over the range of days defined by the slack availability period. The approach followed in this implementation uses a heuristic that modifies the Clark-Wright savings approach [10] for parallel tour generation. A depiction of the general procedure is provided in Figure 3.4.

When this procedure is called by either of the assignment routines, an inspector k is to be assigned a tour to some node i to spend D_k time per visit. For daily assignments, the value D_k is total daily demand remaining, D_r. Thus, whenever D_k+t_k is greater than the inspector's daily time available, a_{kw}, a reduction in the amount of demand that can be satisfied is made within the tour generation procedure. The determination of D_s for non-daily assignments, as explained above, is made at the higher level so D_k is always equal to D_s in this case. In addition to these parameters, the period p, the number of visits, v_{kp}, required of inspector k in period p, and the range of days included in the slack period currently being considered must be known upon initiation.

The next step determines if node i is the depot of inspector k. If it is, no tour is needed. For daily requirements, the inspector can perform inspections up to the amount of daily time available and D_s is determined accordingly. Non-daily assignments have already had D_s determined so the procedure returns immediately.

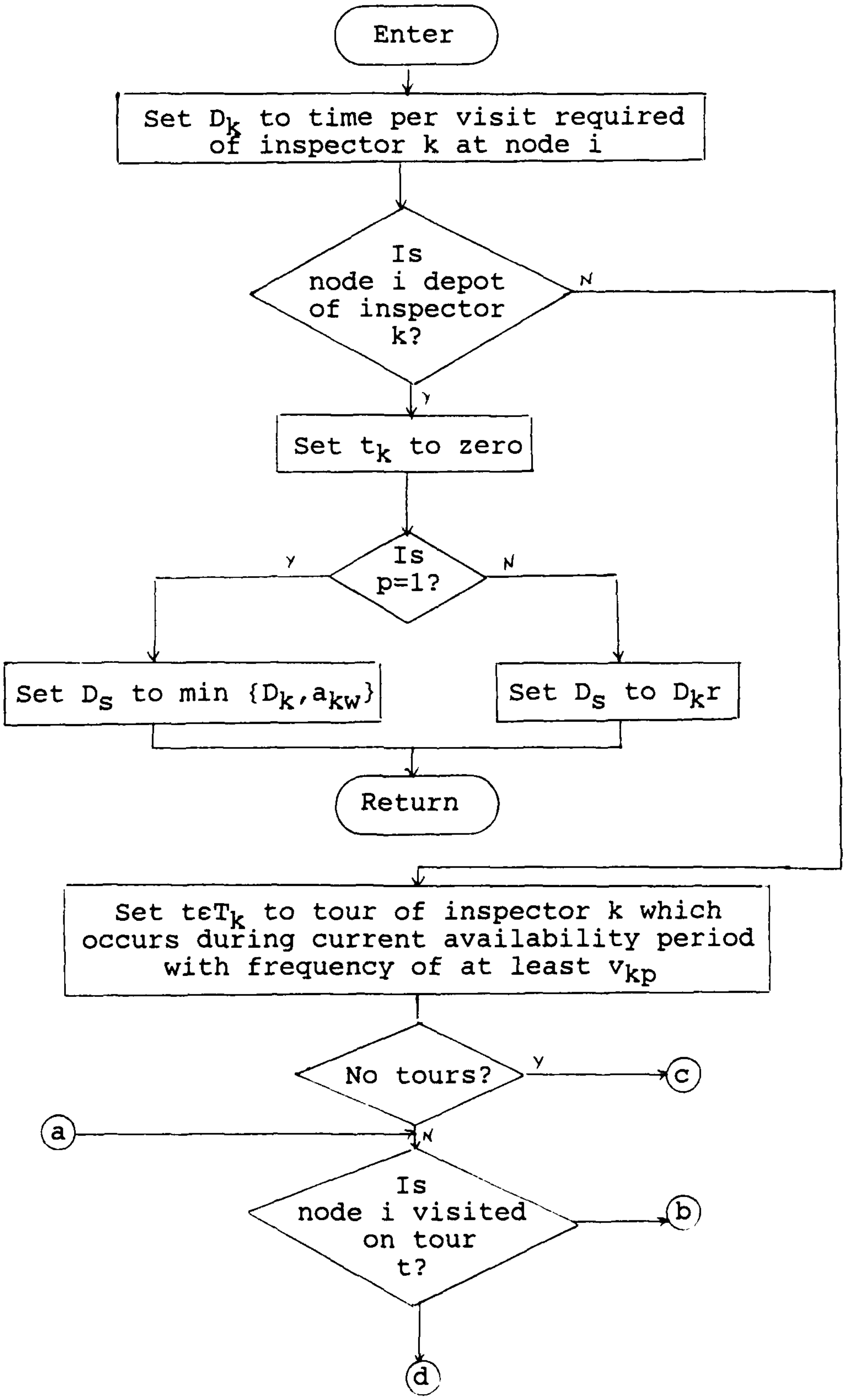


Figure 3.4 Flowchart of Tour Generation Algorithm

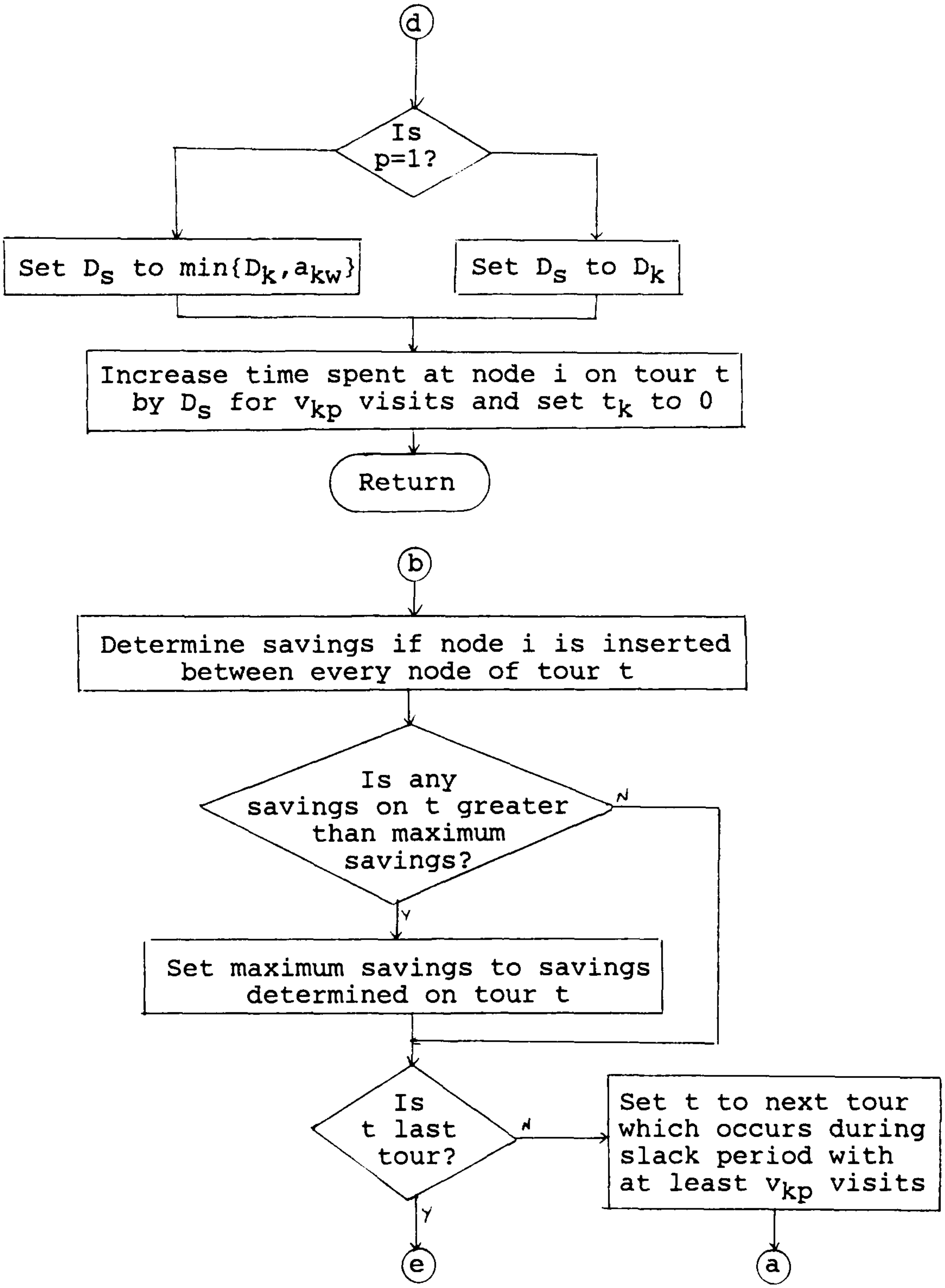


Figure 3.4 (Continued)

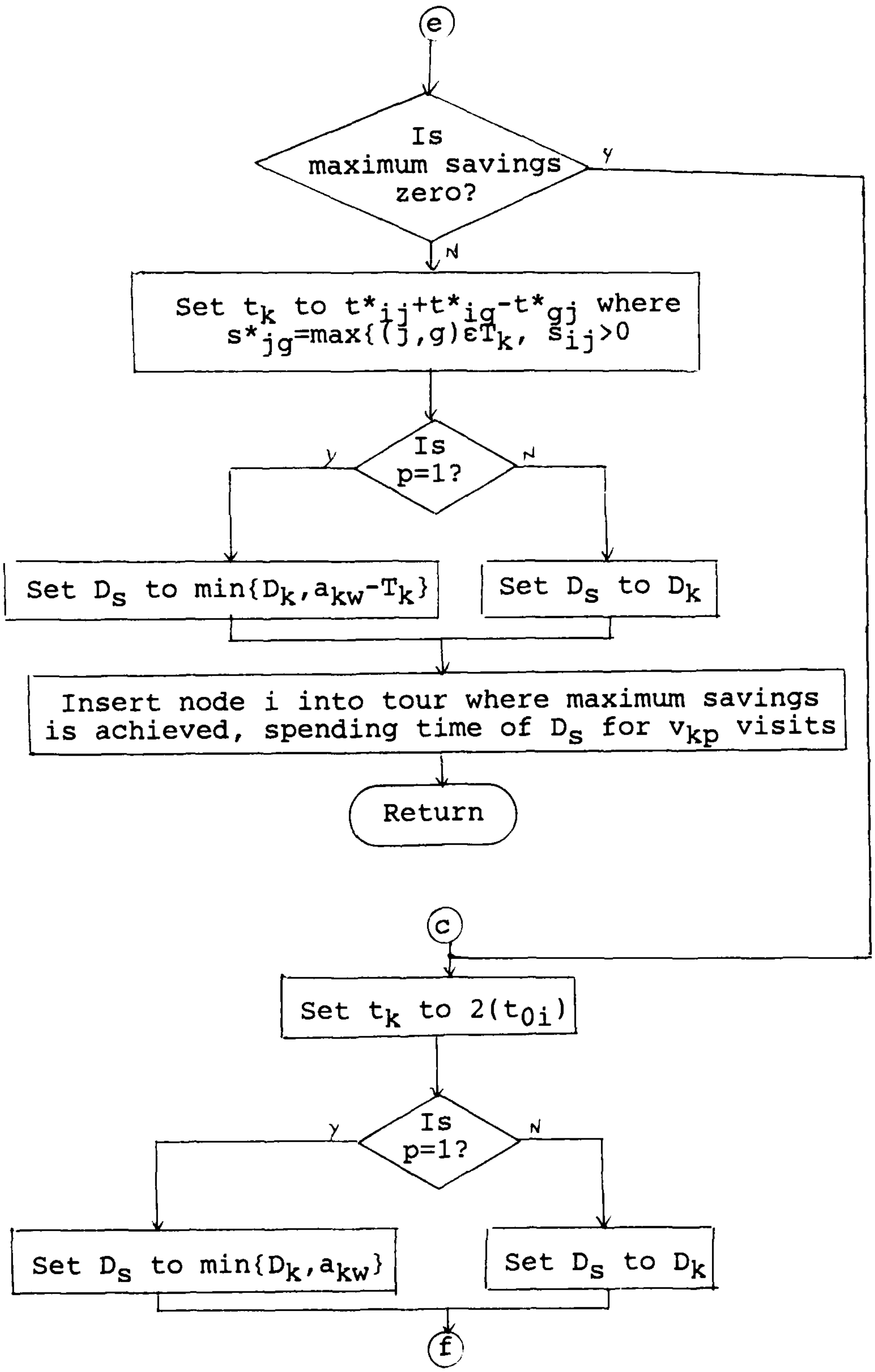


Figure 3.4 (Continued)

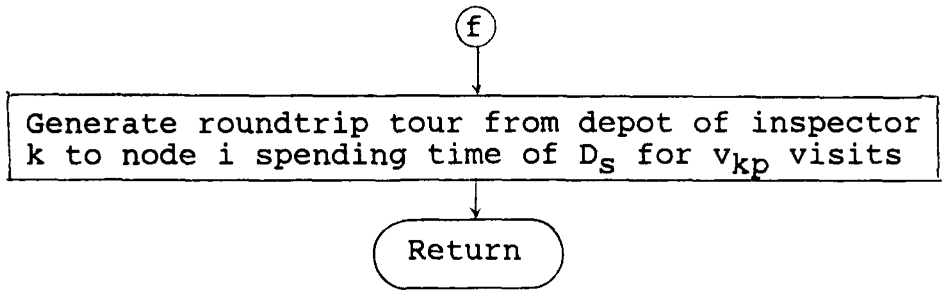


Figure 3.4 (Continued)

When node i is not the inspector's base, a tour assignment is necessary. If the set of all tours of inspector k, T_k, is empty (the inspector has no tours), the shortest distance round-trip tour is generated for the prescribed availability period and number of visits. The index 0 is used to designate an inspector's depot. Again, daily assignments may require an adjustment to the amount of demand satisfied. Otherwise, all of the tours for inspector k which fall within the range of the current slack period and have frequency greater than or equal to the number of visits required (referred to here as feasible tours) are examined to determine: (1) if node i is already visited in one of the inspector's existing tours and (2) where the maximum savings can be achieved if the node is inserted into an existing tour.

Using shortest path data instead of actual arc links, it is possible to visit a node before a demand assignment has been made. Suppose node j is visited by an inspector and a roundtrip tour was generated from the depot. If node i is an intermediate node in the shortest path from the depot to node j, then the actual tour covers node i with no time spent. With no additional

travel time, demand at node i can be satisfied within the inspector's availability and the tour's feasibility constraints.

When the tour does not lie on the path of any existing tour, the tours which fall within the inspector's range of availability are explored to determine where the maximum savings can be achieved. By considering every feasible tour in this manner, the maximum savings is determined over all tours.

The saving between any two nodes j and k is denoted s_{jk} and is defined at the savings which occurs if a node i is inserted between two contiguous nodes on a tour instead of generating a new roundtrip tour from the depot to node i. If we let 0 be the index for the depot, the savings of inserting node i between two nodes j and g is

$$s_{jg} = (t_{ji} + t_{ig} - t_{jg}) - 2t_{0i}.$$

Thus, we find the tour link which provides the best alternative to generating a new roundtrip tour when all the links are examined in every feasible tour.

The feasible tours of an inspector are first examined for the already visited criterion. If the node i is found in one of these tours, T_k is set to zero as no additional travel time is required. For daily demand assignments, the amount of demand that can be satisfied is strictly a function of the inspector's availability as travel time does not impact availability. The amount of demand to be satisfied, as determined for daily assignments or given for non-daily assignments, is added to the amount of time already spent at node i for the required number of

visits.

As each feasible tour is considered, if node i is not found to be already visited, it is considered for insertion between every pair of adjacent nodes of the tour. The saving for each possible insertion is calculated and if it is greater than the current maximum, the maximum saving is updated to reflect the new higher value. In addition, the identity of the tour and the location where the maximum saving occurs are saved. When all feasible tours have been considered under both criteria, the maximum saving is determined.

If this maximum is greater than zero, node i is inserted into the tour between the nodes where the maximum saving was found and T_k is set to the additional travel time to travel to node i. The additional travel time when node i is inserted between nodes j and g is defined as

$$\text{additional travel time} = t_{ji} + t_{ig} - t_{jg}.$$

Again, for daily assignments, the amount of demand that can be satisfied may require reduction. In this case, the amount of demand that can be satisfied is affected by travel time. If the demand required plus the travel time are greater than daily availability, the amount of demand that can be met is daily availability less travel time.

When an existing tour t has a frequency of visits, denoted F_t, that is greater than the number of visits required, v_{kp}, and time spent is increased at a node already visited or a node is inserted into an existing tour, a new tour is essentially formed.

In this case, the nodes and times spent of the original tour are unchanged except for the frequency which is reduced to

$$F_t = F_t - v_{kp}.$$

If the new tour formed with v_{kp} visits required at node i is indexed as tour u, its frequency becomes

$$F_u = v_{kp}.$$

In this way, all nodes of the original tour are visited a total of $(F_t - v_{kp}) + v_{kp} = F_t$ times, the original tour frequency, while node i is visited its required number of times as well.

If all the feasible tours are examined and a maximum saving cannot be achieved from the existing tours, a new roundtrip tour is generated with t_k set to the roundtrip travel time. Here again, since travel time is assigned, the determination of the amount of demand that may be satisfied follows the same approach as if node i was inserted into a tour.

For each depot block, the general approach described above is followed, assigning daily demand first, non-daily demand at nodes visited daily next, and, finally, all remaining nodes with non-daily demand. When all depot blocks have been considered, the procedure terminates. Tour improvement can be applied after each inspector is added and assigned to reduce the overall travel times which in turn increases slack time.

This description illustrates how the daily and non-daily load assignment procedures are linked to tour generation in assigning the minimum number of inspectors. The quality of the tours directly affects the amount of time an inspector is able to

spend performing inspections. To aid in applying this algorithm, a computer program was developed, implementing this basic approach.

3.4 Program Organization

The computer-based implementation of this procedure is organized into eight modules as a FORTRAN program. It was successfully implemented on an IBM PC. The main program module is called ISTAR. The program documentation is included in APPENDIX A and a complete program listing is provided in APPENDIX B. Figure 3.5 depicts the organizational structure of the various modules. The modules are consolidations of major program elements which allow for convenient changes if modified approaches or different heuristics are desired.

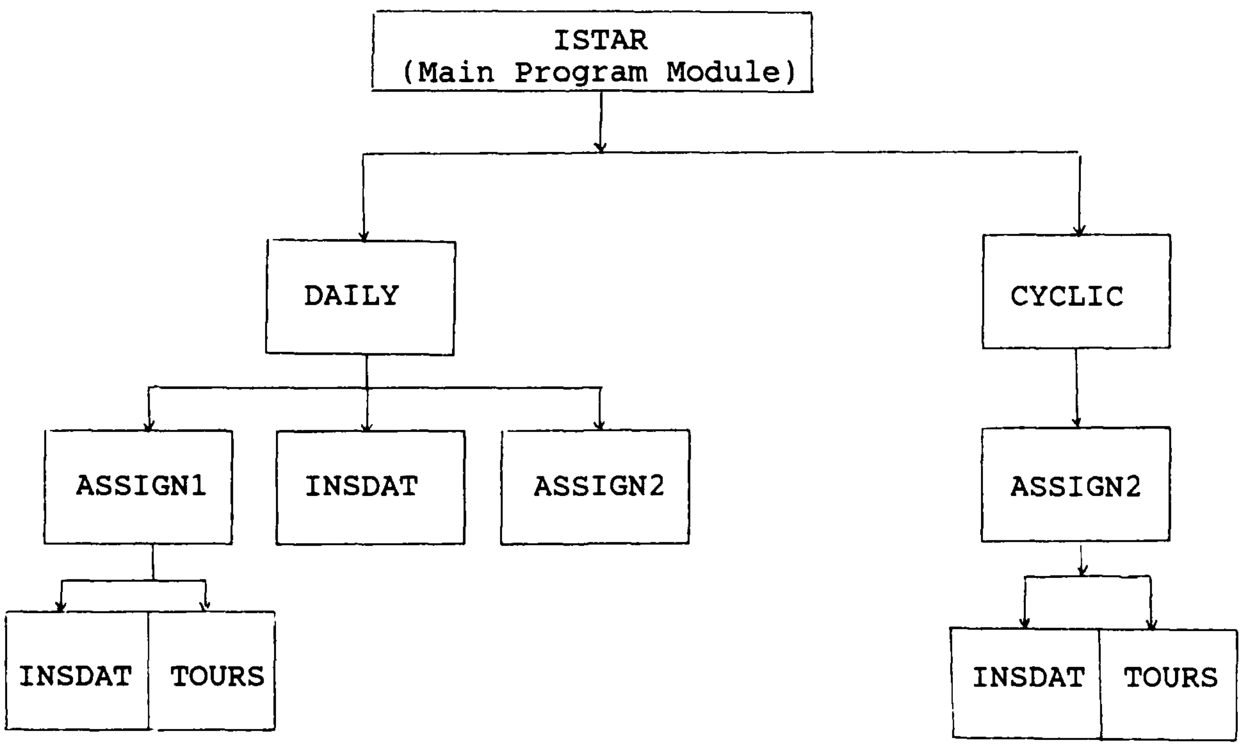


Figure 3.5 Hierarchy of Program Modules

The main algorithm presented in the last section and shown in Figure 3.1 is implemented by the main program module called ISTAR. The purpose of this module is to initialize the given parameters, read network and demand data, and keep track of the block of nodes assigned to each depot for demand satisfaction. Calls are then made to the DAILY and CYCLIC modules for the assignment of daily and non-daily demands respectively for each block. After all depots are completed, the OUTPUT module is called to provide file output of the final slack summary for each inspector, the tours for each inspector, the assignments made for each node, and the utilization statistics.

The assignment of daily demand is controlled by the DAILY program module. Nodes of the current block with positive daily demand are assigned inspectors through calls to the ASSIGN1 module. All daily demand for a given node is satisfied in the ASSIGN1 module. The INSDAT module is then called to find inspectors with daily slack availability over the total working period. A record is kept of all nodes visited with daily demand and those with positive non-daily demands are assigned inspectors through successive calls to ASSIGN2. Thus, upon return from the DAILY module, the nodes from a given depot block with positive daily demand have not only their daily demands satisfied but their non-daily demands are met as well.

The control of assignments for the nodes with only positive non-daily demand (no daily demand) is accomplished by the CYCLIC module. Starting with the highest non-daily frequency period the

demand for these nodes is satisfied through calls to the ASSIGN2 module. A record is kept of all nodes visited for each frequency period so that prior to stepping to the next lower period, all the lower frequency demand of the nodes visited is satisfied first. Returning from the CYCLIC module, all non-daily demands for the depot block are satisfied.

The INSDAT module is called to analyze the inspector slack data and provide as many inspectors as are needed with adjacent slack periods to cover the entire number of working days. If total coverage cannot be achieved, additional inspectors are added until complete coverage is achieved. From the slack time per day and the number of days the slack period covers, each inspector's equivalent number of days available to perform inspections under each frequency class is calculated. The portion of this module which adds an inspector is called every time there is a need for a new inspector. In this way, inspectors are added such that a new inspector's first day of availability occurs immediately after the last day of the previous inspector added and ensures continuity between the daily and non-daily assignment routines.

The actual assignment of an inspector to satisfy daily demand is accomplished in the ASSIGN1 module. This segment implements the daily assignment procedure shown in Figure 3.2. Here, all daily demand is satisfied over the total working period for a given node. Inspectors with slack are handled singly in an attempt to use as much slack time as possible to satisfy demand.

If no inspectors have slack for the period needed, new inspectors are added. The precise amount of demand satisfied by a given inspector is determined through calls to the TOURS module which returns the demand that can be satisfied given that a specific tour is used. The daily slack of each inspector assigned is then reduced to reflect the inspection and travel times used for daily assignments.

Similarly, the ASSIGN2 module implements the non-daily assignment procedure depicted in Figure 3.3, assigning inspectors to satisfy non-daily demand. Working with an inspector or inspectors who have combined daily slack coverage of the entire working period, an attempt is made to consume as much of their slack time as possible to satisfy non-daily demand for a given node. Except for the fact that this module often deals with multiple inspectors and the non-daily demands must be converted to daily equivalents, the method parallels that of ASSIGN1 to accommodate multiple depots. The exact amount of demand satisfied is again determined by calls to the TOURS module. When slack for an inspector becomes insufficient, a call is made to the INSDAT module to provide another set of inspectors where inspectors are added if needed.

The TOURS module assigns tours to all inspectors determined to have sufficient slack time by the ASSIGN1 and ASSIGN2 modules. This module implements the tour generation algorithm of Figure 3.4. Using shortest distance path data, tours are created using optimal travel times. Before creating a new tour, existing tours

of an inspector are explored to determine if a node is already visited. If not, they are examined further to determine where the maximum savings can be achieved if the node is inserted between any two nodes of the inspector's existing tours.

Finally, the OUTPUT module contains six independent subroutines to perform file output. Subroutine TROUT provides a listing of all the tours generated for each inspector. IDATOUT provides the inspector data output which is generated in the INSDAT module. The ADOUT subroutine provides a final listing of all assignments made with total demand satisfied by node and frequency period. SLKOUT is a subroutine to output a tabular summary of all inspectors' slack time per day organized over the periods for which it is constant. This subroutine is called after assignments are made for different frequency periods to show the results after significant iterations. Subroutine LOCOUT provides a listing of the most recent assignments made to locations of a particular frequency period. Lastly, STATS compiles pertinent demand statistics and calculates individual and overall inspector utilization rates.

4. APPLICATION

4.1 An Example

This section describes how the proposed algorithm is applied to a demand data set. The test data is from an example detailed by Dessouky et al. [12] to provide a basis for comparison. In this way, we see the similarity between the two approaches as well as how the results are affected by the differences presented in the description of Chapter 3.

The first step of the algorithm in Figure 3.1 calls for the initialization of all parameters. The network depicted in Figure 4.1 has 19 demand nodes to be visited over a period of a year. Using the algorithm by Floyd, the shortest times between every pair of nodes in the network are determined. Table 4.1 represents these travel times, t_{ij}, between any two nodes i and j. The shortest chain matrix of intermediate nodes which coincides with these times is presented in Table 4.2.

The problem considers a total of seven frequency periods covering a time period of one year. It is determined that there is a total of W working days requiring inspections during the year (W = 252). Inspectors are available to perform inspections a maximum of A days per year (A = 218) due to programmed vacation time. Further, an inspector is considered to have a maximum of a_{kw} hours per day to perform inspections (initially $a_{kw} = 6.5$) for each inspector k.

For an annual time f`ame, the seven frequency periods are defined in Table 4.3. Th the value of P is seven and f_p

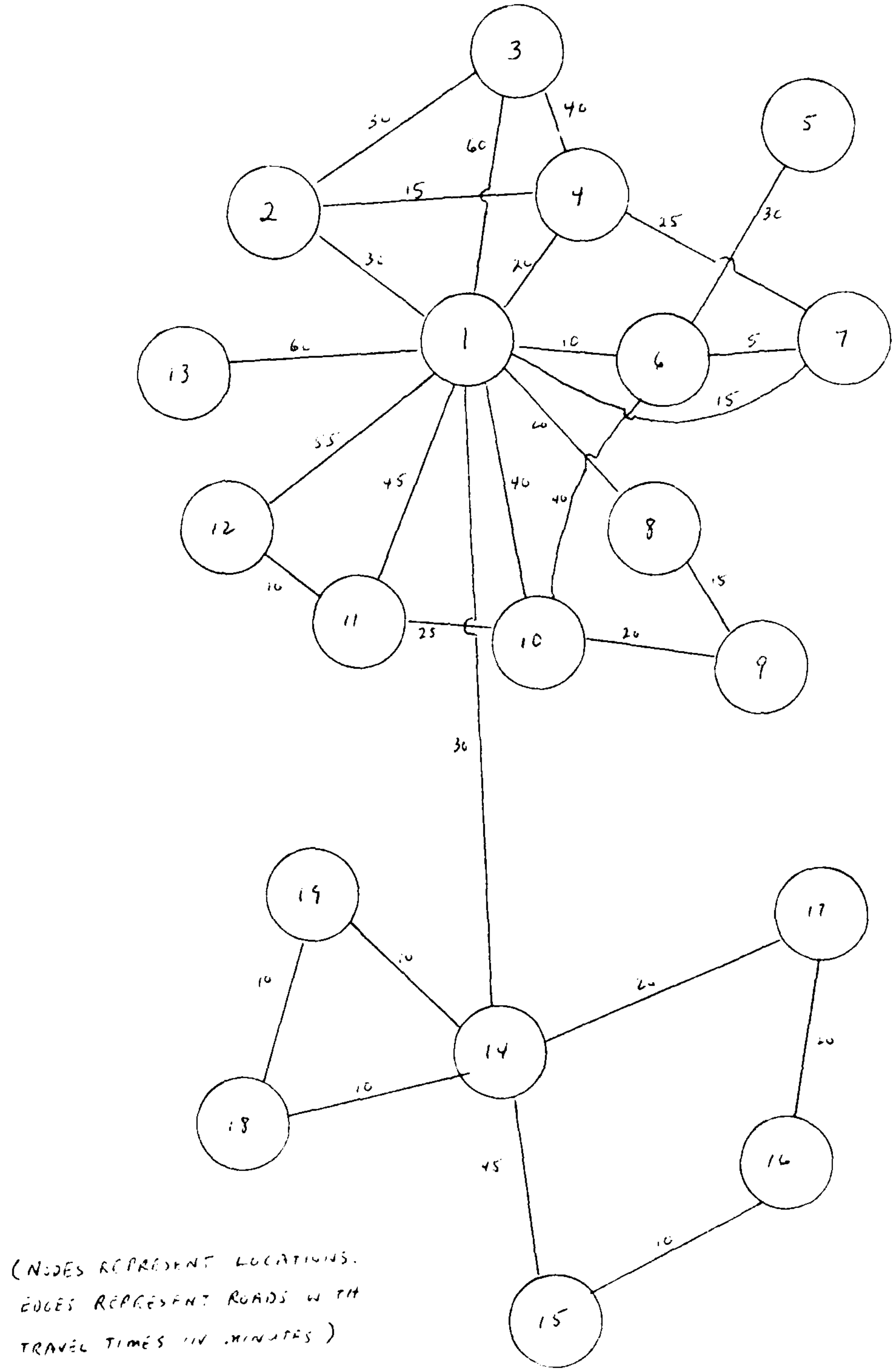


Figure 4.1 Network of Inspection Nodes and Travel Routes

Table 4.1 Shortest Times Between Nodes

	1	2	3	4	5	6	7	8	9	10
1	0	30	60	20	40	10	15	20	35	40
2	30	0	30	15	65	35	40	50	65	70
3	60	30	0	40	90	60	65	80	95	100
4	20	15	40	0	50	20	25	40	55	60
5	40	65	90	50	0	30	35	60	75	70
6	10	35	60	20	30	0	5	30	45	40
7	15	40	65	25	35	5	0	35	50	45
8	20	50	80	40	60	30	35	0	15	35
9	35	65	95	55	75	45	50	15	0	20
10	40	70	100	60	70	40	45	35	20	0
11	45	75	105	65	85	55	60	60	45	25
12	55	85	115	75	95	65	70	70	55	35
13	60	90	120	80	100	70	75	80	95	100
14	30	60	90	50	70	40	45	50	65	70
15	70	100	130	90	110	80	85	90	105	110
16	60	90	120	80	100	70	75	80	95	100
17	50	80	110	70	90	60	65	70	85	90
18	40	70	100	60	80	50	55	60	75	80
19	40	70	100	60	80	50	55	60	75	80

	11	12	13	14	15	16	17	18	19
1	45	55	60	30	70	60	50	40	40
2	75	85	90	60	100	90	80	70	70
3	105	115	120	90	130	120	110	100	100
4	65	75	80	50	90	80	70	60	60
5	85	95	100	70	110	100	90	80	80
6	55	65	70	40	80	70	60	50	50
7	60	70	75	45	85	75	65	55	55
8	60	70	80	50	90	80	70	60	60
9	45	55	95	65	105	95	85	75	75
10	25	35	100	70	110	100	90	80	80
11	0	10	105	75	115	105	95	85	85
12	10	0	115	85	125	115	105	95	95
13	105	115	0	90	130	120	110	100	100
14	75	85	90	0	40	30	20	10	10
15	115	125	130	40	0	10	30	50	50
16	105	115	120	30	10	0	20	40	40
17	95	105	110	20	30	20	0	30	30
18	85	95	100	10	50	40	30	0	10
19	85	95	100	10	50	40	30	10	0

Table 4.2 Shortest Chain Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1	2	3	4	6	6	7	8	8	10	11	12	13	14	14	14	14	14	14
2	1	2	3	4	4	4	4	1	1	1	1	1	1	1	1	1	1	1	1
3	1	2	3	4	4	4	4	1	1	1	1	1	1	1	1	1	1	1	1
4	1	2	3	4	6	6	7	1	1	1	1	1	1	1	1	1	1	1	1
5	6	6	6	6	5	6	6	6	6	6	6	6	6	6	6	6	6	6	6
6	1	4	4	4	5	6	7	1	1	10	1	1	1	1	1	1	1	1	1
7	1	4	4	4	6	6	7	1	1	6	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	8	9	9	9	9	1	1	1	1	1	1	1
9	8	8	8	8	8	8	8	8	9	10	10	10	8	8	8	8	8	8	8
10	1	1	1	1	6	6	6	9	9	10	11	11	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	10	10	10	11	12	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	11	11	11	11	12	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	13	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	14	16	16	17	18	19
15	16	16	16	16	16	16	16	16	16	16	16	16	16	16	15	16	16	16	16
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	15	16	17	14	14
17	14	14	14	14	14	14	14	14	14	14	14	14	14	14	16	16	17	14	14
18	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	18	19
19	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	18	19

Table 4.3 Inspection Frequencies

Frequency Class (p)	Inspection Frequency	f_p (Visits/Year)
1	Daily	252
2	Weekly	52
3	Bi-weekly	26
4	Monthly	12
5	Quarterly	4
6	Semi-annually	2
7	Annual	1

denotes the frequency of visits required in each frequency period p.

These frequency classes then give rise to the demand information which is included as Data Set 1 in APPENDIX C. The demand summaries are broken into inspection time required for

each visit and the total demand required to be satisfied over the total working period of 252 days. The times required per visit, d_{ip}, represent the amount of time an inspector is needed at node i during period p. The annual demand summary is the total inspection requirements for the year.

This data, when supplied to the computer program implementing the algorithm presented in Chapter 3, yielded output which is included as APPENDIX D. The remainder of this section will describe how the algorithm is applied to achieve this output.

The second step in Figure 3.1 calls for determining the depots and assigning nodes to each depot. For this problem, this information is taken as given. Two depots are located at nodes 1 and 14. Nodes 2 through 13 and nodes 15 to 19 are assigned to the two depots respectively. Thus, the first depot block is identified as nodes 1 through 13 and we proceed to the next step of assigning inspectors to fulfil daily demand requirements at these nodes.

Two nodes from this block require inspectors on a daily basis, nodes 1 and 6, with requirements of 8.3 and 1.6 hours per visit respectively. The daily demand iterations follow the procedure outlined in Figure 3.2. Daily demand assignments begin with node 1 so D_r is set to 8.3. Also, w is set to one, the first working day. Since there are initially no inspectors, inspector 1 is assigned (indexed by k = 1), based at the current depot (node 1), with availability of 218 days from day 1 to 218.

Next, the amount of demand that can be satisfied by inspector 1 is determined using the procedure of Figure 3.4. Applying this procedure, it is determined that since inspector 1 is based at node 1, no tour is needed ($t_1 = 0$) and the amount of demand that can be met is given by

$$D_s = \min\{D_r, a_{1w}\} = \min\{8.3, 6.5\} = 6.5.$$

Thus, D_r is reduced by the amount satisfied, leaving 8.3 - 6.5 = 1.8 hours remaining. Inspector 1 is then assigned to meet 6.5 hours per day at node 1 for the first 218 days and a_{1w} for each day w = 1,...,218 is reduced to zero. For day 218, w reaches the last day of the availability period for inspector 1. Since we have not reached the number of working days of 252, w is set to 219, and an inspector with availability for day 219 and beyond is sought. Again, no inspector is found so inspector 2 is added with availability beginning where inspector 1 left off. Thus, inspector 2 is available for 218 days from day 219 through 252 and day 1 to 184. The procedure assigns the same amount of demand to inspector 2 from days 219 to 252 leaving no availability for this period. At this point, w reaches the total number of working days but D_r is still 1.8 hours so the procedure is restarted at day 1.

Now inspector 2 is found with 6.5 hours per day available from days 1 to 184. Inspector 2 is also based at node 1 so no tour is needed. In this case, D_r is less than the daily availability of inspector 2 so

$$D_s = \min\{D_r, a_{kw}\} = \min\{1.8, 6.5\} = 1.8$$

and inspector 2 is assigned to meet 1.8 hours per day at node 1 from days 1 to 184 and daily availability for this period is reduced to 6.5 - 1.8 = 4.7 hours per day. D_r is now reduced to zero. At this point w is set to day 185 and another inspector with availability is sought to cover the remaining time from days 185 to 252. Since no inspector currently assigned has availability during this period, inspector 3 is added with availability beginning on day 185. Inspector 3 is assigned 1.8 hours from day 185 to 252 which completes the assignment of daily demand at node 1. This time w represents day 252 and the demand remaining, D_r, is zero so the procedure terminates.

Table 4.4 provides a summary of the remaining time available for each of the three inspectors after daily demand at node 1 is fulfilled. Figure 4.2 shows how the daily demand at node 1 is met by the three inspectors and Figure 4.3 illustrates their inspection loads.

Table 4.4 Summary of Availability Periods

	Period(Days)	Available Days	Hrs/Day
Inspector 1:	1 - 252	0	0.00
Inspector 2:	1 - 184	184	4.70
	185 - 252	0	0.00
Inspector 3:	1 - 150	150	6.50
	151 - 184	0	0.00
	185 - 252	68	4.70

The daily assignment procedure is then applied to meet the demand, D_r, of 1.6 hours per visit at node 6. The day w is set to the first working day. Inspector 2 is determined to have

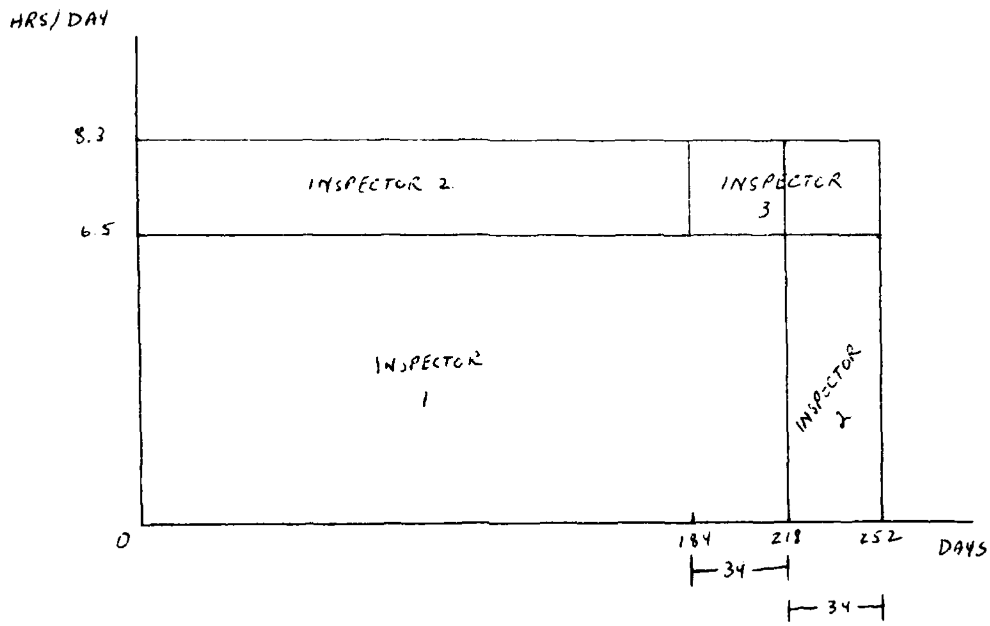


Figure 4.2 Inspection Load Profile for Node 1

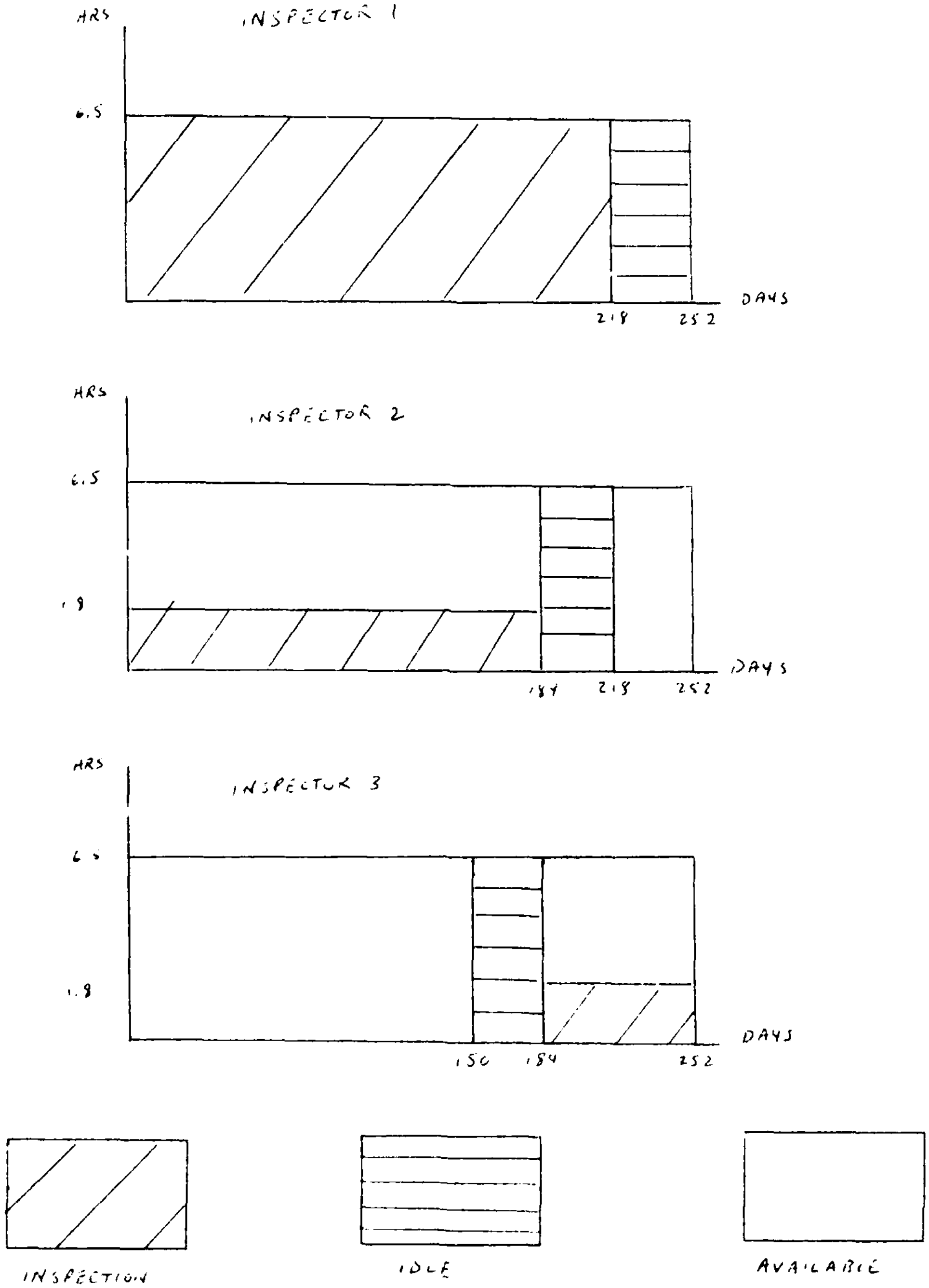


Figure 4.3 Individual Inspector Load Profiles for Daily Inspections at Node 1

sufficient slack time during the first 184 days if the time to travel to node 6 is less than the daily time available of 4.7 hours. In this case, no tour exists for inspector 2 so the roundtrip travel time of

$$t_2 = 2(10) = 20 \text{ minutes; } 0.33 \text{ hours}$$

applies. Thus, inspector 2 has sufficient availability. The generation of this first tour (1 - 6 - 1) and the determination of the amount of demand to be satisfied are again accomplished by the tour generation procedure. In this case, the amount of demand inspector 2 may meet is given by

$$D_s = \min\{D_r, a_{kw}-T_k\} = \min\{1.6, 4.7-0.33\} = 1.6 \text{ hours.}$$

Inspector 2 is assigned to meet daily demand at node 6 using this tour on each of the first 184 days. To account for the 1.8 hours already assigned at node 1, this tour is limited to 4.7 hours. Daily time available is reduced to

$$a_{2w} = 4.7 - (1.8 + 0.33) = 2.77 \text{ hours per day}$$

for $w = 1,\ldots,184$.

The same approach is followed for the remaining period from day 185 to 252 using inspector 3. Another roundtrip tour is generated for inspector 3 covering the last 68 days of the working period with the same amount of demand satisfied. Thus, daily time available for inspector 3 is also

$$a_{3w} = 4.4 - (1.8 + 0.33) = 2.77 \text{ hours per day}$$

for $w = 185,\ldots,252$. The updated load profiles are shown graphically in Figure 4.4 for inspectors 2 and 3. Table 4.5 gives the updated availabilities.

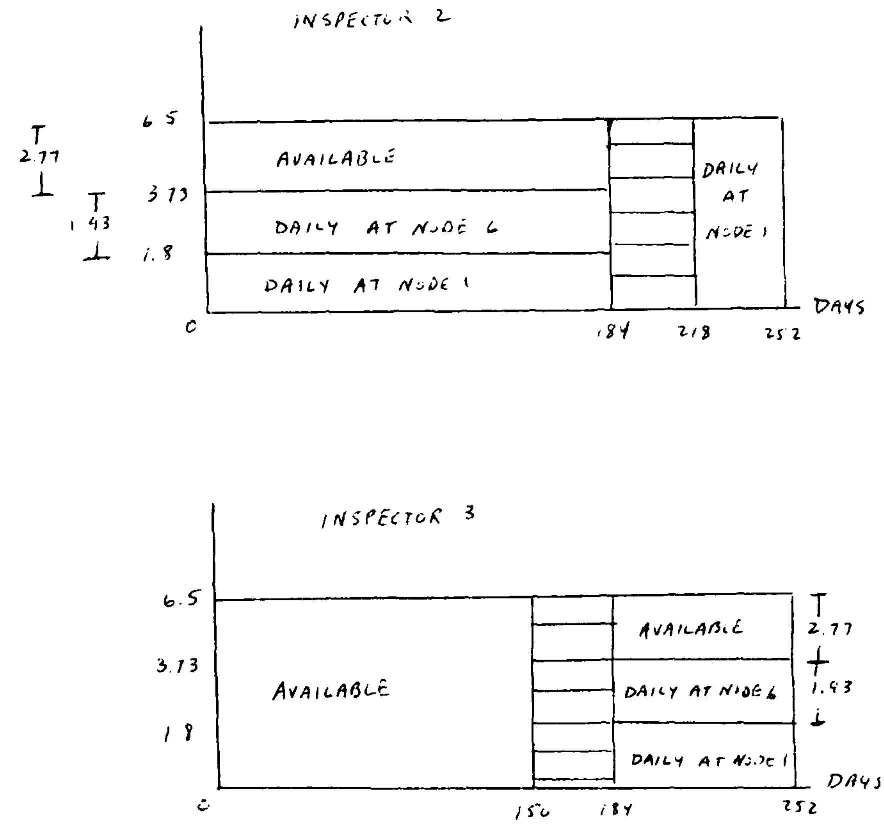


Figure 4.4 Load Profiles of Inspectors 2 and 3 for Daily Inspections at Nodes 1 and 6

Table 4.5 Summary of Availability Periods

	Period(Days)	Available Days	Hrs/Day
Inspector 1:	1 - 252	0	0.00
Inspector 2:	1 - 184	184	2.77
	185 - 252	0	0.00
Inspector 3:	1 - 150	150	6.50
	151 - 184	0	0.00
	185 - 252	68	2.77

Continuing with the next step of the algorithm, the non-daily demands at these two nodes are satisfied before the remaining nodes requiring non-daily visits are considered. To accomplish this task, the non-daily assignment procedure given in Figure 3.3 is applied to nodes 1 and 6 starting with the highest non-daily frequency period and working to the lowest.

Before non-daily assignments can be made, a set of inspectors with slack coverage over the total working period must be obtained. From Table 4.5, inspectors 2 and 3 provide complete coverage of all working days with periods from day 1 to 184 and day 185 to 252 respectively.

Once a set of inspectors with complete coverage is obtained, the members are ordered so that the inspector whose availability period provides the greatest number of total hours is considered first. We then calculate the number of class p visits, v_{kp}, each inspector k of the set may perform. If d_k denotes the number of days in the availability period of inspector k, f_p denotes the frequency of period p visits, W is the total number of working days, and r_p is the number of remaining days of class p visits

that must be covered, then v_{kp} is given by

$$v_{kp} = \min\{[d_k f_p/W],\ r_p\}$$

such that [z] denotes the nearest integer greater than or equal to z. Initially, r_p is set equal to f_p and is updated to

$$r_p = r_p - v_{kp}$$

after the coverage of each inspector k of the set is determined.

Hence from the example, the 184 days of inspector 2, whose total hours of availability for the 184 day period is greater than that of inspector 3, may be used to cover:

(1) min{[(184)52/252=37.96],52} = 38 weekly inspections

(2) min{[(184)26/252=18.9],26} = 19 bi-weekly inspections

(3) min{[(184)12/252=8.7],12} = 9 monthly inspections

(4) min{[(184)4/252=2.9],4} = 3 quarterly inspections

(5) min{[(184)2/252=1.46],2} = 2 semi-annual inspections

(6) min{[(184)1/252=0.73],1} = 1 annual inspection.

The remaining days of visits required are then updated as follows:

(1) For p = 2, r_p = 52 - 38 = 14 days

(2) For p = 3, r_p = 26 - 19 = 7 days

(3) For p = 4, r_p = 12 - 9 = 3 days

(4) For p = 5, r_p = 4 - 3 = 1 day

(5) For p = 6, r_p = 2 - 2 = 0 days

(6) For p = 7, r_p = 1 - 1 = 0 days.

The coverage of inspector 3 is calculated in the same manner as inspector 2. In this case, the value of $d_k f_p/W$ is greater than r_p for all periods p so v_{3p} receives the values of r_p above,

representing the coverage for inspector 3. These results are included in the output as the equivalent availabilities in days by frequency class.

Since we start with the highest non-daily frequency period, we consider node 6 first with weekly demand of 0.5 hours. The next step is to determine the travel time t_k for each inspector k of the set. Since node 6 is visited daily by both inspectors 2 and 3 during the availability periods being considered (and node 1 is their depot) no additional travel time is required and their respective t_k is zero.

The next step examines how much of the weekly demand at node 6 inspectors 2 and 3 can fulfil based on their remaining slack time. Here, we again define S_k as the total time available of inspector k in the current availability period of length d_k days with a_{kw} time remaining per day. S_k is thus given by

$$S_k = a_{kw}d_k.$$

The amount of time required for an inspector k with T_k additional travel time to perform v_{kp} visits is given by

$$\text{total time required to travel} = t_k v_{kp}.$$

Letting $Dmax_{sk}$ denote the maximum amount of demand that can be satisfied by an inspector k. The procedure starts by first checking whether the additional travel time (if any) can be accommodated within that slack amount. This implies

$$Dmax_{sk}v_{kp} = S_k - t_k v_{kp} \geq 0.$$

If more time is available for inspections, the amount of demand fulfilled by an inspector k, D_{sk}, will be equal to

$$D_{sk} = \min \{Dmax_{sk}, D_r\}.$$

For the weekly demand at node 6, inspector 2 has total availability of

$$S_k = (184)(2.77) = 509.68 \text{ hours}$$

and inspector 3 has

$$S_k = (68)(2.77) = 188.36 \text{ hours.}$$

The total amount of time required to meet the demand of D_r is given by

$$\text{time required} = D_r v_{kp}.$$

Thus, the time required for inspector 2 is (0.5)(38) = 19.0 hours and the time required for inspector 3 is (0.5)(14) = 7.0 hours. Since these amounts are less than the their total available time in the period, all of the demand may be satisfied and D_{sk} is set to 0.5 hours for each inspector. D_s, the demand to be satisfied is then defined as

$$D_s = \min \{D_{sk}\}$$

for all inspectors k in the set. Here, all weekly demand for node 6 is supplied in one iteration.

The actual assignment includes the time of 0.5 hours per visit being assigned to both inspectors for their required number of visits during their respective availability periods. The amount of each inspector's total availability, S_k, is then decreased by the total time required giving

$$S_k = S_k - (D_s + T_k) v_{kp}.$$

For inspector 2, availability in the period is reduced by (0.5)(38) = 19.0 hours leaving 509.7 - 19.0 = 490.68 hours

remaining. Likewise, inspector 3 uses (0.5)(14) = 7.0 hours which reduces total availability to 188.4 - 7.0 = 181.4 hours.

The same set of inspectors have sufficient slack to handle all of the remaining non-daily demands for these two nodes as demonstrated in Table 4.6. Each assignment causes a reduction in slack time which is reflected by the decrease in total number of hours remaining in the period.

Table 4.6 Summary of Non-daily Assignments to Nodes Visited Daily

Coverage
Inspector 2: Days 1 - 184
Inspector 3: Days 185 - 252

p	f_p	Node	Demand	Travel	Inspector	Total hours Required	Total hours Remaining
2	52	6	0.5	0.0	2	(0.5)(38)=19.0	509.7
					3	(0.5)(14)=7.0	188.4
4	12	1	8.7	0.0	3	(8.7)(9)=78.3	490.7
						(8.7)(3)=26.1	181.4
		6	4.6	0.0	2	(4.6)(9)=41.4	412.4
					3	(4.6)(3)=13.8	155.3
5	4	1	6.0	0.0	2	(6.0)(3)=18.0	371.0
					3	(6.0)(1)=6.0	141.5
		6	3.2	0.0	2	(3.2)(3)=9.6	353.0
					3	(3.2)(1)=3.2	135.5
6	2	1	0.9	0.0	2	(0.9)(2)=1.8	343.4
		6	3.9	0.0	2	(3.9)(2)=7.8	341.6
7	1	1	4.8	0.0	2	(4.8)(1)=4.8	333.8
		6	1.4	0.0	2	(1.4)(1)=1.4	329.0

Upon completing the last of the above assignments, inspector 2 is left with 329.0 - 1.4 = 327.6 hours and inspector 3 has

135.5 - 3.2 = 132.3 hours remaining. This equates to an average of 327.6/184 = 1.78 hours per day for inspector 2 and 132.3/68 = 1.94 hours per day for inspector 3 in their respective availability periods which is how the output included in APPENDIX D records this information.

The methodology used by Dessouky et al. maintains a record of the slack days remaining, in determining whether to assign an inspector. The equivalent number of slack days used is calculated as follows:

$$\text{number of days} = (D_s + t_k)v_{kp}/a_{kw}$$

where a_{kw} is the daily availability for some day w in the current availability period. Table 4.7 shows the results of their computations for the assignment of non-daily demand for nodes 1 and 6. For inspector 2, the total number of slack days used in the period sums to 65.81 days so 184 - 65.73 = 118.27 days with 2.77 hours of slack time remaining in the period. Likewise, inspector 3 is left with 68 - 20.24 = 47.76 days remaining in the period.

The two approaches are shown to be essentially equivalent by comparing the total number of slack hours left in the slack period for each inspector. For inspector 2,

$$(118.27)(2.77) \approx 327 \text{ total hours} \approx (1.78)(184)$$

and for inspector 3,

$$(47.76)(2.77) \approx 132 \text{ total hours} \approx (1.94)(68).$$

Using the same periods of availability for inspectors 2 and 3, the remaining nodes with demand are satisfied. Starting with

the weekly frequency period, nodes 4 and 11 have requirements. The summary of these iterations are shown in Table 4.8. These nodes have not been visited previously so tours must be generated with travel time incorporated into the availability computations.

Table 4.7 Computation of Slack Days [12]

Period	Node	Demand	Equivalent Number of Slack Days: Inspector 2	Equivalent Number of Slack Days: Inspector 3
2	6	0.5	$\frac{(0.5)(38)}{2.77}=6.86$	$\frac{(0.5)(14)}{2.77}=2.53$
4	1	8.7	$\frac{(8.7)(9)}{2.77}=28.27$	$\frac{(8.7)(9)}{2.77}=9.42$
	6	4.6	$\frac{(4.6)(9)}{2.77}=14.94$	$\frac{(4.6)(3)}{2.77}3=4.98$
5	1	6.0	$\frac{(6.0)(3)}{2.77}=6.50$	$\frac{(6.0)(1)}{2.77}=2.16$
	6	3.2	$\frac{(3.2)(3)}{2.77}=3.46$	$\frac{(3.2)(1)}{2.77}=1.15$
6	1	0.9	$\frac{(0.9)(2)}{2.77}=0.65$	
	6	3.9	$\frac{(3.9)(2)}{2.77}=2.82$	
7	1	4.8	$\frac{(4.8)(1)}{2.77}=1.73$	
	6	1.4	$\frac{(1.4)(1)}{2.77}=0.50$	
			Σ = 65.73 days	Σ = 20.24 days

Node 4, with 0.8 hours of demand, is handled first. The time required to travel to node 4 for each of the two inspectors is determined to be $t_{14} + t_{41} - t_{16} = 20 + 20 - 10 = 30$ minutes (0.5 hours) which is the additional travel time when node 4 is

added to each of the inspector's (1-6-1) round-trip tours. This is determined by the savings achieved when the following comparison is made:

(1) (1 - 6 - 1) = 20 minutes travel

(1 - 4 - 1) = 40 minutes travel => 60 minutes total

and (2) (1 - 4 - 6 - 1) = 50 minutes total

for a savings of 10 minutes. Inspector 2, with 38 required visits, is therefore routed from node 1 to node 4 before node 6 a total of 38 times during the availability period.

Table 4.8 Summary of Non-daily Assignments for Nodes Visited Weekly

Coverage
Inspector 2: Days 1 - 184
Inspector 3: Days 185 - 252

						Total Hours	
p	f_p	Node	Demand	Travel	Inspector	Required	Remaining
2	52	4	0.8	0.5	2	(0.5+0.8)(38)=49.4	327.6
					3	(0.5+0.8)(14)=18.2	132.3
		11	0.6	1.5	2	(1.5+0.6)(38)=79.8	278.2
					3	(1.5+0.6)(14)=29.4	114.1
4	12	4	2.6	0.0	2	(2.6)(9)=23.4	198.4
					3	(2.6)(3)=7.8	84.7
		11	0.8	0.0	2	(0.8)(9)=7.2	175.0
					3	(0.8)(3)=2.4	76.9
6	2	4	2.1	0.0	2	(2.1)(2)=4.2	170.8
7	1	4	0.4	0.0	2	(0.4)(1)=0.4	166.6
		11	2.4	0.0	2	(2.4)(1)=2.4	166.2

Node 11 requirements are met by generating a new round-trip tour (1 - 11 - 1) because there is no savings possible, given the

current set of tours. Tour 3 is thus added for both inspectors.

The lower frequency demand periods for the two nodes visited weekly are assigned in the same manner as for the nodes visited daily. No additional travel time is required for either inspector visiting these nodes during lower frequency periods. These results are also shown in Table 4.8. The total time required to meet all demands for these nodes leaves inspectors 2 and 3 with 163.8 and 74.5 hours remaining in their respective availability periods.

Nodes with monthly demand are considered next. Table 4.9 shows the non-daily assignment summary for the four locations with monthly demand which are nodes 2, 3, 7, and 13.

Node 2 is assigned first. The maximum savings occurs when node 2 is inserted into the inspectors' (1 - 6 - 1) tour. Since the shortest distance from node 2 to 6 is via node 4, the resulting tour is (1 - 2 - 4 - 6 - 1). The resulting savings is 35 minutes with additional travel time of $t_{12} + t_{26} - t^{1}{}_{6} = 30 + 35 - 10 = 55$ minutes (0.92 hours). Inspector 2 is therefore routed a total of nine times to node 2 prior to visiting node 6. Inspector 3 uses the same tour three times.

Node 3 is the next node with monthly demand. The maximum savings possible occurs when node 3 is also visited before node 6 on the roundtrip tour. The additional travel time is $t_{13} + t_{36} - t_{16} = 60 + 60 - 10 = 110$ minutes (1.83 hours). The resulting tour is (1 - 3 - 4 - 6 - 1) with frequency of 9 and 3 days for the two inspectors respectively.

The next node with monthly demand is node 7. The maximum savings is determined to occur when node 7 is visited before node 6 on the round-trip tour between the depot and node 6. The additional travel time is $t_{17} + t_{76} - t_{16} = 15 + 5 - 10 = 10$ minutes (0.17 hours). The last node with monthly demand, node 13, requires a new round-trip tour as no savings is found by insertion into the existing tours. Thus, the travel time is 120 minutes (2.00 hours) and the tour (1 - 13 - 1) is generated for each inspector.

Table 4.9 Summary of Non-daily Assignments for Nodes Visited Monthly

Coverage
Inspector 2: Days 1 - 184
Inspector 3: Days 185 - 252

p	f_p	Node	Demand	Travel	Inspector	Total Hours Required	Total Hours Remaining
4	12	2	1.2	0.92	2	(1.2+0.92)(9)=19.1	163.8
					3	(1.2+0.92)(3)=6.4	74.5
		3	0.6	1.83	2	(0.6+1.83)(9)=21.9	144.7
					3	(0.6+1.83)(3)=7.3	68.1
		7	2.6	0.17	2	(2.6+0.17)(9)=24.9	122.8
					3	(2.6+0.17)(3)=8.3	60.8
		13	1.5	2.0	2	(1.5+2.0)(9)=31.5	97.9
					3	(1.5+2.0)(3)=10.5	50.3
7	1	2	0.7	0.0	2	(0.7)(1)=0.7	66.4
		3	2.3	0.0	2	(2.3)(1)=2.3	65.7
		7	0.3	0.0	2	(0.3)(1)=0.3	63.4
		13	0.6	0.0	2	(0.6)(1)=0.6	63.1

Updating the total time remaining for each inspector after

the last monthly node is assigned, we have inspector 2 with 62.5 hours and inspector 3 with 39.8 hours.

The only node of the depot block having demand that remains unvisited is node 12 with yearly demand of 0.40 hours. This demand is met by routing inspector 2 through node 12 prior to node 11 once a year forming the tour (1 - 12 - 11- 1). The additional travel time of $t_{1\ 12} + t_{12\ 11} - t_{1\ 11} = 55 + 10 - 45 =$ 20 minutes (0.33) hours. Inspector 2 meets the required demand of (1.2 + 0.33)(1) = 1.5 hours leaving 61.0 hours remaining in the availability period. This equates to the 0.32 hours per day for inspector 2 and the 39.8 hours of inspector 3 are equivalent to 0.61 hours per day.

At this point the procedure outlined above is repeated for the second depot block, nodes 14 through 19. However, before proceeding, this procedure is improved by looking ahead and forecasting the need for another inspector. The method adopted takes the total time remaining of the inspectors who have been added so far

$$\text{total availability of inspectors} = \sum_{k=1}^{n} \sum_{w=1}^{W} a_{kw}$$

and compares this time with the amount of demand to be fulfilled at the nodes of the next depot block given by

$$\text{total demand of depot block} = \sum_{i \varepsilon D_i} \sum_{p=1}^{P} d_{ip} f_p$$

where D_i denotes the set of all nodes assigned to the ith depot. If the total demand exceeds the total slack time of the

inspectors, another inspector is added, based at the new depot.

An inspector added in this situation receives priority in the daily assignment procedure. This allows the higher frequency visits to be assigned to a locally based inspector before calling on inspectors from other depots. Inspectors from other depots are "saved" for the lower frequency periods at other depots. Since their travel time includes the time to reach the depot plus travel to a node of the block, better utilization of slack time is achieved and the approach is intuitively more reasonable.

For the problem at hand, the total amount of slack time for the three inspectors added thus far is:

(1) 61.0 hours for inspector 2 over days 1 to 184

(2) 39.8 hours for inspector 3 over days 185 to 252

(3) (6.5)(150) = 975 hours for inspector 3 over days 1 to 150. Their total availability sums to 1075.8 hours. The demand for nodes 14 through 19 totals 1443.3 hours so a fourth inspector is added.

The daily assignment procedure is now applied as before to the two nodes with daily demand, making maximum use of the new fourth inspector based at node 14. Using all of the 218 days that inspector 4 is available, only 34 days are required from inspector 3. Both inspectors have all of their daily availability of 6.5 hours per day in the periods assigned. A summary of the daily assignments at this depot block are shown in Table 4.10. Inspector 3 is required to travel to node 14, forming the roundtrip tour (1 - 14 - 1) with time of 60 minutes

(1.0 hour). The additional travel time for inspector 3 and the roundtrip travel time for inspector 4 to node 19 is 20 minutes (0.33 hours). These assignments are shown in Table 4.10.

The assignment of non-daily demand is accomplished in the same manner as for the first depot. The results of the non-daily assignments for the two nodes visited daily are shown in Table 4.11. Table 4.12 concludes the remaining non-daily assignments for the second depot. After the assignment of monthly demand for node 15, inspector 4 is left with 1.3 hours remaining on the first 116 days. This amount is insufficient to meet the requirements for node 18. Inspector 3 has 6.5 hours per day remaining for the first 116 days, replaces inspector 4 to provide complete coverage of all days, and another inspector is not needed. The algorithm terminates after this second depot is completed, determining that four inspectors are required.

Table 4.10 Summary of Daily Assignments for Second Depot

Node	Inspector	Demand	Travel	Days	Total Hours Required	Total Hours Remaining
14	4	2.0	[illegible]	1-116	232.0	754.0
	3	2.0	[illegible]	117-150	102.0	221.0
	4	2.0	[illegible]	151-252	204.0	663.0
19	4	0.8	0.33	1-116	131.1	522.0
	3	0.8	0.33	117-150	38.4	119.0
	4	0.8	0.33	151-252	115.2	459.0

Dessouky et al. also conclude a minimum of four inspectors are required. The lower bound on the number of inspectors for this problem is determined to be

lower bound = [(total demand)/(total availability)].

Table 4.11 Summary of Non-daily Assignments for Nodes of Second Depot Visited Daily

Node	Inspector	Visits	Demand	Travel	Days	Total Hours Required	Total Hours Remaining
14	4	24	0.7	0.0	1-116	16.8	390.9
	4	22	0.7	0.0	151-252	15.4	343.8
	3	6	0.7	0.0	117-150	4.2	80.6
		52					
19	4	24	2.5	0.0	1-116	60.0	374.1
	4	22	2.5	0.0	151-252	55.0	328.4
	3	6	2.5	0.0	117-150	15.0	76.4
		52					
14	4	6	7.7	0.0	1-116	46.2	314.1
	4	5	7.7	0.0	151-252	38.5	273.4
	3	1	7.7	0.0	117-150	7.7	61.4
		12					
19	4	6	15.2	0.0	1-116	91.2	267.9
	4	5	15.2	0.0	151-252	76.0	234.9
	3	1	15.2	0.0	117-150	15.2	53.7
		12					
19	4	2	1.2	0.0	1-116	2.4	176.7
	4	2	1.2	0.0	151-252	2.4	158.9
		4					
14	4	1	1.2	0.0	1-116	1.2	174.3
	4	1	1.2	0.0	151-252	1.2	156.5
		2					
14	4	1	4.9	0.0	1-116	4.9	173.1
19	4	1	8.4	0.0	1-116	8.4	168.2

Table 4.12 Summary of Non-daily Assignments for Remaining Nodes

Node	Inspector	Visits	Demand	Travel	Days	Total Required	Hours Remaining
16	4	24	0.6	1.0	1-116	38.4	159.8
	4	22	0.6	1.0	151-252	35.2	155.3
	3	6	0.6	1.0	117-150	9.6	38.5
		52					
17	4	24	0.8	0.17	1-116	23.3	121.4
	4	22	0.8	0.17	151-252	21.3	120.1
	3	6	0.8	0.67	117-150	8.8	28.9
		52					
16	4	6	1.4	0.0	1-116	8.4	98.1
	4	5	1.4	0.0	151-252	7.0	98.8
	3	1	1.4	0.0	117-150	1.4	20.1
		12					
17	4	6	13.8	0.0	1-116	82.8	89.7
	4	5	13.8	0.0	151-252	69.0	91.8
	3	1	13.8	0.0	117-150	13.8	18.7
		12					
15	4	6	0.6	0.33	1-116	5.6	6.9
	4	5	0.6	0.33	151-252	4.6	22.8
	3	1	0.6	1.33	117-150	1.9	4.9
		12					
18	3	6	1.0	1.33	1-116	14.0	754.0
	4	5	1.0	0.17	151-252	5.8	18.2
	3	1	1.0	0.17	117-150	1.2	3.0
		12					
18	3	1	0.6	0.0	1-116	0.6	740.0

For this data set, the lower bound is

$$\lceil 4372.0/1417.0 \rceil = \lceil 3.09 \rceil = 4 \text{ inspectors.}$$

Therefore, this algorithm has determined the true minimum number of inspectors.

4.2 Daily Demand Extremes

The algorithm was next evaluated using the same network as in the example above with the same amounts of non-daily demand but with daily demand now required at every node.

Overall algorithm performance is expected to be reduced from the previous example. A total of 19 inspectors are required to meet this new demand schedule using a total of 84 tours. Overall utilization of time for inspections averages 53.96% for the inspectors and their time spent traveling averages 29.51%.

The magnitudes of the daily demands ensure that travel time will consume a large percentage of the inspectors' time available. By assigning all daily demands first, many of the inspectors who visit nodes daily have their slack times used up before the nodes they visit are considered for non-daily demand satisfaction. Thus, this case represents an extreme problem applied to the algorithm to demonstrate its application and performance beyond the scope for which it was initially devised.

The algorithm is effective at using inspector slack time as 13 of the 19 inspectors are assigned to use more than 85% of their total available time. Also, if a similar slack consolidation is made, the last two inspectors assigned are used only 7.76% and 36.18% of their total available time which is less

than the availability of two other inspectors. In this case, one may conclude that 17 inspectors are required, 6 more than the lower bound of 11.

The opposite case was then applied to the algorithm, this time with the same non-daily demands but with no daily demand. Using a total of 218 working days, we know that a minimum of two inspectors are required to satisfy demand over the total of 252 working days despite the fact that the lower bound computes to one for this case. Without completely using all of the slack times of two inspectors, a total of 18 tours are generated to meet all the demand in this problem. Inspector 1 is utilized for 96.73% of total availability of which 75.07% is time spent inspecting.

Again, the problem shows the diversity of the algorithm to problems beyond which is was intended. In this case, performance proves to be optimal. Clearly, the higher frequency demands, particularly daily demand, provide the greatest challenge for the method.

4.3 A Four-depot Network

As a final demonstration of the applicability of the program, a simple four-depot network was devised with daily demand required at each node. The network is shown in Figure 4.5 with depots at nodes 1, 5, 9, and 13. The demand requirements are listed in Table 4.10. This problem demonstrates the effectiveness of the modifications made to the methodology of Dessouky et al in successfully handling multiple depots.

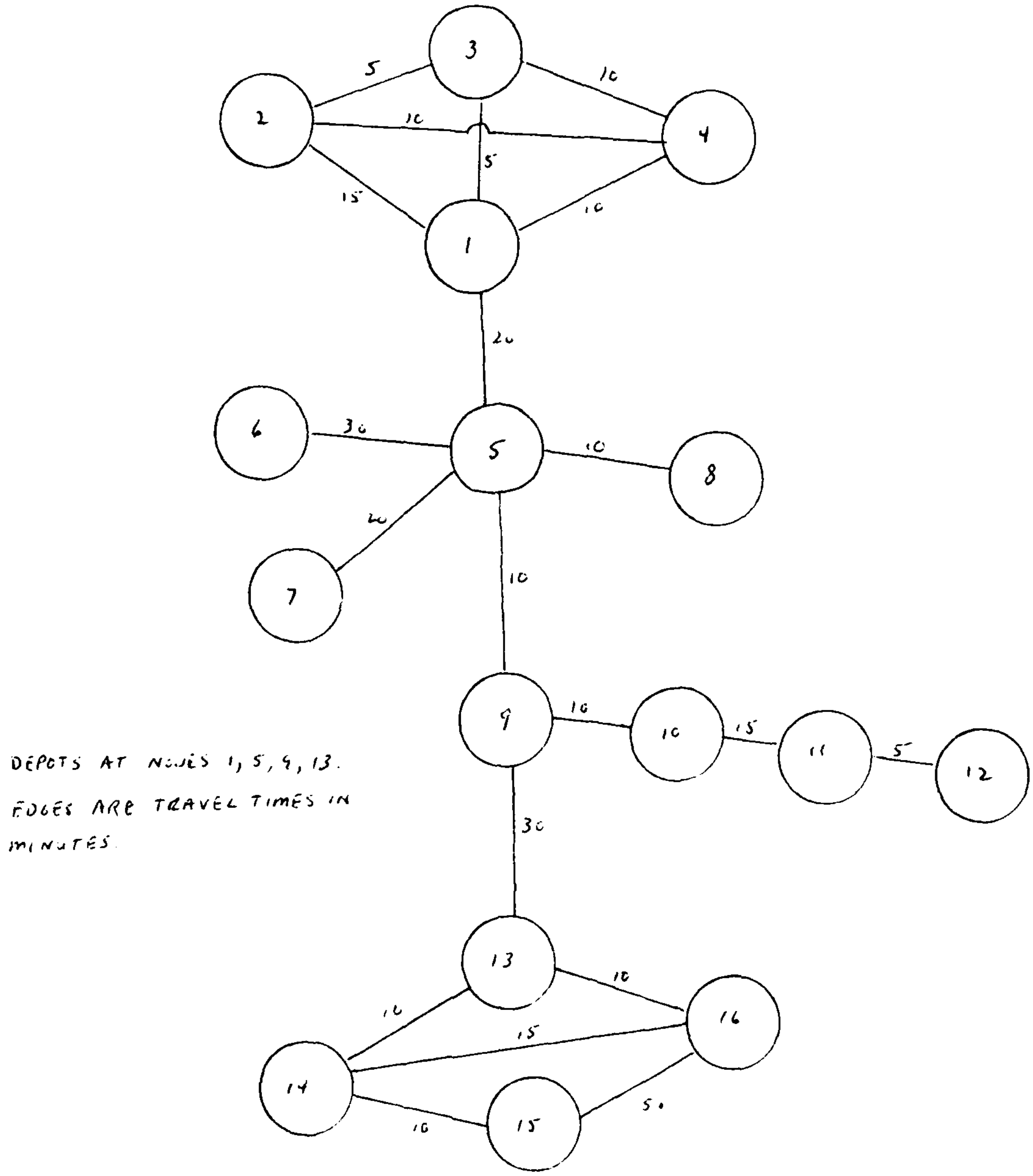


Figure 4.5 Four-depot Network

The lower bound on the number of inspectors required to meet this demand is eight. A total of 13 inspectors are assigned with 35 tours generated. Seven of the inspectors are used over at least 85% of their total availability with the time assigned to the last two inspectors greater than the slack remaining for several other inspectors.

Table 4.13 Demands of Four-depot Network

	Frequency Period						
Node	1	2	3	4	5	6	7
1	8.3	0	0	8.7	6.0	0.9	4.8
2	2.0	0	0	1.2	0	0	0.7
3	1.2	0	0	0.6	0	0	2.3
4	1.0	0.8	0	2.6	0	2.1	0.4
5	6.2	0	0	0	0	0	0
6	1.6	0.5	0	4.6	3.2	3.9	1.4
7	0.3	0	0	2.6	0	0	0.3
8	4.1	0	0	0	0	0	0
9	2.0	0	0	0	0	0	0
10	0.1	0	0	0	0	0	0
11	8.5	0.6	0	0.8	0	0	2.4
12	1.4	0	0	0	0	0	0.4
13	2.6	0	0	1.5	0	0	0.6
14	2.0	0.7	0	7.7	0	1.2	4.9
15	0.5	0	0	0.6	0	0	0
16	0.7	0.6	0	1.4	0	0	0

The first five inspectors are assigned to the depot at node 1. The first three of these have utilization of virtually all of their slack time and time spent inspecting of 100.00, 92.43, and 88.27% as their duties are primarily associated with nodes of their own depots. The last two inspectors at the first depot have dramatic increases in time spent traveling to approximately 40% with virtually all slack time being consumed. This results from the added travel time to nodes outside of their depot block.

The same trend occurs for the inspectors assigned to the second and third depots as well, however, their utilization of time spent inspecting is significantly reduced. This occurs because inspectors from other depots are assigned to the higher frequency periods at other depots before new inspectors are added to the depot. When added, they are used for lower frequency visits to nodes in their depot block.

While not incorporated in these applications, this trend indicates the need for some limit being established for inspectors traveling to satisfy demand at nodes beyond their own depot. Taken to an extreme, it is possible for an inspector to travel a large period to meet some small amount of demand when inspectors at the local depot are available to meet the same demand more efficiently.

Recognizing this problem, the approach of adding inspectors as each new depot is encountered and assigning them to the highest frequency visits was added. This method effectively prevents the extreme example above and serves to maximize utilization of the inspectors by reducing overall travel time.

5. PARAMETRIC ANALYSIS

5.1 Determining Number of Inspectors

The first twelve data sets of APPENDIX C represent the demand required at the 19 nodes of Figure 4.1 for separate skill categories. The data was compiled by Dessouky et al. for determining the number of inspectors required to meet demand by individual skill category. Skill categories could be defined as air conditioning inspectors, building inspectors, water inspectors, and so on. The important point to note is that demand for each skill is different and each category is treated as a separate problem.

In this section, the required number of inspectors determined by the proposed algorithm are compared to the results obtained by Dessouky et al. for each of the separate data sets (skill categories). The input parameters are the same as for the example presented in Chapter 4: the same seven frequency periods, total working days of 252, maximum inspector time available per day of 6.5 hours, and depots located at nodes 1 and 14. Separate comparisons were made using 218 and 252 days of inspector availability.

The separate demand data sets were input to the computer program listed in APPENDIX B with the results shown in Table 5.1. The comparison shows that in most cases the algorithm determined an added inspector over what was calculated by Dessouky et al. Two differences in the approaches followed account for this disparity. First, Dessouky et al. allow an unassigned inspector

from the first depot to be based at the second depot for the period of time over which they are unassigned. This decreases the overall travel time required in their computations. Secondly, since they use a single depot formulation of the inspector allocation problem, after all assignments are completed for the first depot block, they assign additional inspectors to the second depot by comparing remaining demand load to existing inspector availability. If total demand remaining for the second block is greater than total inspector availability, an additional inspector or inspectors are needed. Assignment of demand is not made by the inspectors' availability periods.

Table 5.1 Required Number of Inspectors (218 Working Days)

	Minimum Number of Inspectors		
Data Set	Dessouky[12]	Program	Lower Bound
1	4	4	4
2	3	4	3
3	4	4	4
4	4	5	4
5	3	3	3
6	3	4	4
7	2	3	2
8	2	3	2
9	3	4	3
10	2	2	2
11	2	3	2
12	-	2	2

Only in four out of the twelve data sets does the proposed algorithm achieve the lower bound minimum number of inspectors. In the other eight cases, an additional inspector is required using this algorithm when compared to the results of Dessouky et al. For data set 12, Dessouky et al. assign no inspectors to

this trivial case with extremely low demand. However, the true minimum for this demand set is two inspectors because total inspector availability is between 50% and 100% of the total number of working days and complete coverage is required to meet even this limited demand schedule. The program performs as designed for this case of low demand.

The same data sets were then applied to the program with the total number of days of inspector availability increased to the total required of 252 days. The results are summarized in Table 5.2. Here we see the program achieves the lower bound in every set but data set 11. For data set 12, Dessouky et al. assign no inspectors to this trivial case but the lower bound is one, which the algorithm determines. For data set one, the program achieves the lower bound estimate of three inspectors, one less inspector than by the method of Dessouky et al.

Table 5.2 Required Number of Inspectors (252 Working Days)

	Number of Inspectors		
Data Set	Dessouky[12]	Program	Lower Bound
1	4	3	3
2	3	3	3
3	3	3	3
4	4	4	4
5	3	3	3
6	3	3	3
7	2	2	2
8	2	2	2
9	3	3	3
10	1	1	1
11	1	2	1
12	-	1	1

The algorithm demonstrates superior performance in determining the required number of inspectors when inspectors are allowed 252 days of availability. This fact suggests that some improvement is needed when the total number of working days is greater than the number of days of inspector availability. The approach adopted was to find the inspectors with the greatest daily availability for each working day and give them priority in non-daily assignments.

Various experiments were then conducted to determine how often this priority scheme should be applied to ensure that inspectors with the greatest availability are assigned first. It was found that the update produced the best results when applied once at the beginning of the assignment of non-daily demand for nodes visited daily and once at the start of assigning non-daily demand to the remaining nodes for each depot block. In this way, an inspector who visits a node for the first time in a higher frequency period is afforded the maximum opportunity to revisit the node in lower frequency periods without incurring more travel time. The results from applying this modification are given in Table 5.3. Here, six of the twelve data sets achieve the lower bound estimate reached by Dessouky et al. This modification allowed the algorithm to achieve its best performance.

The results summarized in this section show that the assignment approach of Dessouky et al. produces superior results over the 218 day inspector availability period. When the working days is equal to the number of days available, the algorithm

Table 5.3 Required Number of Inspectors Using Modification (218 Working Days)

	Minimum Number of Inspectors		
Data Set	Dessouky[12]	Program	Lower Bound
1	4	4	4
2	3	4	3
3	4	4	4
4	4	5	4
5	3	3	3
6	3	3	3
7	2	3	2
8	2	3	2
9	3	4	3
10	2	2	2
11	2	3	2
12	-	2	2

produced the same results. When availability of the resource in question is similar to the assignment heuristics used in this algorithm (such as aircraft with periodic maintenance requirements covering an extended period of time), the algorithm can still be of use in making good estimates of the number of resources required.

5.2 Skills Combined

In this section, the demands of the individual skill categories are combined to evaluate the algorithm performance over a wide range of demand loads. Intuitively, we expect the number of inspectors to be reduced from the requirements of the last section when separate demand data is combined to form higher-level skill categories. Increased demand loading should increase the amount of demand assigned to inspectors for each node visited, resulting in more efficient use of their slack time. Consequently, utilization rates for time spent inspecting

are expected to improve as well.

The result of combining individual skills on the number of inspectors required is shown in Table 5.4. The data sets were grouped as depicted. The total number of inspector days available was 218. Under the column for number of inspectors required when the skills are treated separately, the data from the previous section is used. The first number is the total given by Dessouky et al and the second is the number determined by the computer program.

Combining data sets in this manner in effect results in the forming of a new, higher-level skill category with inspectors capable of handling multiple types of inspections. The number of inspectors required becomes less and less as more skill groups are combined together.

The same combination of skill groups were then compared using 252 days of inspector availability. The summary of results are shown in Table 5.5. Here again, we see the same improvement in overall number of inspectors when skills are combined.

If all of the individual skills are combined to a single, high-level skill category, one can expect to reduce the total number of inspectors required to 26 for 218 days availability and 22 for 252 days of availability. This represents an improvement of from 6 to 18 and 7 to 9 inspectors respectively, depending on the what method is used for determining the individual number required. The actual reduction can then be compared with the increased cost of inspectors with multiple skills to predict

Table 5.4 Number of Inspectors with Skills Combined (218 Working Days)

Data Sets Combined	Number of Inspectors: Separate Skills Dessouky[12]	Separate Skills Program	Skills Combined
1 + 2	7	9	7
3 + 4	8	9	9
5 + 6	6	7	6
7 + 8	4	6	4
9 + 10	5	6	5
11 + 12	2	5	3
1 + 2 + 3 + 4	15	18	14
5 + 6 + 7 + 8	10	13	9
9 + 10 + 11 + 12	7	11	6
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8	25	31	21
Total of 1 - 12	32	42	26

Table 5.5 Number of Inspectors with Skills Combined (252 Working Days)

Data Sets Combined	Number of Inspectors: Separate Skills Dessouky[12]	Separate Skills Program	Skills Combined
1 + 2	7	6	6
3 + 4	7	7	7
5 + 6	6	6	5
7 + 8	4	4	3
9 + 10	4	4	4
11 + 12	1	3	2
1 + 2 + 3 + 4	14	13	12
5 + 6 + 7 + 8	10	10	7
9 + 10 + 11 + 12	5	7	5
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8	24	23	18
Total of 1 -12	29	31	22

Table 5.6 Utilization as a Function of Total Demand

Data Set(s)	Total Demand (Hrs)	Time Spent Inspecting
1	4372.00	61.71%
2	3571.00	63.00%
3	4369.80	61.68%
4	5125.60	72.34%
5	3187.80	56.24%
6	3469.80	61.22%
7	1603.80	37.73%
8	2108.60	49.60%
9	3487.00	61.52%
10	1110.10	39.17%
11	1557.50	36.64%
12	149.80	5.29%
1,2	7943.00	80.08%
3,4	9495.40	74.46%
5,6	6657.60	78.31%
7,8	3712.40	71.27%
9,10	4597.10	64.88%
11,12	1707.30	40.16%
1 - 4	17348.40	87.90%
5 - 8	10370.00	81.31%
9 - 12	6304.40	74.15%
1 - 8	27808.40	93.45%
1 - 12	34112.80	92.59%

whether a net savings is possible.

Having seen the reductions that are possible in the number of inspectors required, we focus next on how the improvement translates to improved inspector utilization as a function of demand load. Here, the emphasis will be on utilization of time spent inspecting rather than use of slack time. The total annual demand loads from the individual as well as combined data sets are given in Table 5.6 with the percentage time spent inspecting for all inspectors assigned. Letting q_{kw} denote the time an inspector k spends on day w performing inspections (not including travel times), the utilization of all inspectors time for inspections is

$$\text{utilization} = \frac{\sum_{k=1}^{n} \sum_{w=1}^{W} q_{kw}}{\sum_{k=1}^{n} \sum_{w=1}^{W} a^*_{kw}}$$

where a^*_{kw} is the maximum daily time available for any inspector k on day w, W is the total number of working days, and n is the total number of inspectors from the inspector allocation formulation of Chapter 1. Results are given for total inspector availability of 218 days.

The results of Table 5.6 show at a glance that the average time an inspector spends performing inspections is improved significantly with increased demand loads. At the low extreme, data set 12 results in only 5.29% utilization. On the opposite end, the combination of sets one through eight provides the highest utilization of 93.45%.

The relationship between demand load and utilization is best illustrated by Figure 5.1. Through combining data sets, the wide range of demand loads show the tendency to level-off at a high level of utilization. Exactly where the plateau occurs is, of course, a function of the number of nodes, the travel times, and the quality of the tour generating method employed. For this problem, it appears to be around 90%.

In this section, it was demonstrated how this algorithm can be used to optimize resources. As demand load increases, overall inspector utilization is increased to a certain point where it then reaches an upper limit. When separate skills are combined to achieve the increased demand, there is a corresponding

reduction in the number of inspectors, just as expected.

5.3 Optimizing Utilization

The precise points from Table 5.6 which characterize the curve of Figure 5.1 are scattered around the line. Note, for example, that utilization is higher for demand when the first eight skills are combined than it is when all of the skills are combined. This suggests that utilization may actually cycle regularly above and below the curve. If the precise nature of the utilization curve is understood, the information can be useful in determining demand amounts which achieve the highest levels of utilization.

To gain this understanding, the demand of data set one is again used as a reference. This data set is then divided into increments of tenths. Starting with one-tenth of the original demand, the data is applied to the program to analyze how many inspectors are assigned and their overall time spent inspecting. Continuing with two-tenths up to 1.9 times the original data set, the number of inspectors is tracked along with utilization rates in an effort to evaluate the precise nature of the curve of Figure 5.1. The results of these runs are shown in Table 5.7.

We note immediately the trend which is exhibited in the summary of Table 5.7. When demand reaches a certain point, the number of inspectors is increased and utilization falls. A graphical representation of this data is included as Figure 5.2. Cycles around the characteristic curve continue with each subsequent peak and valley being higher than the last until the

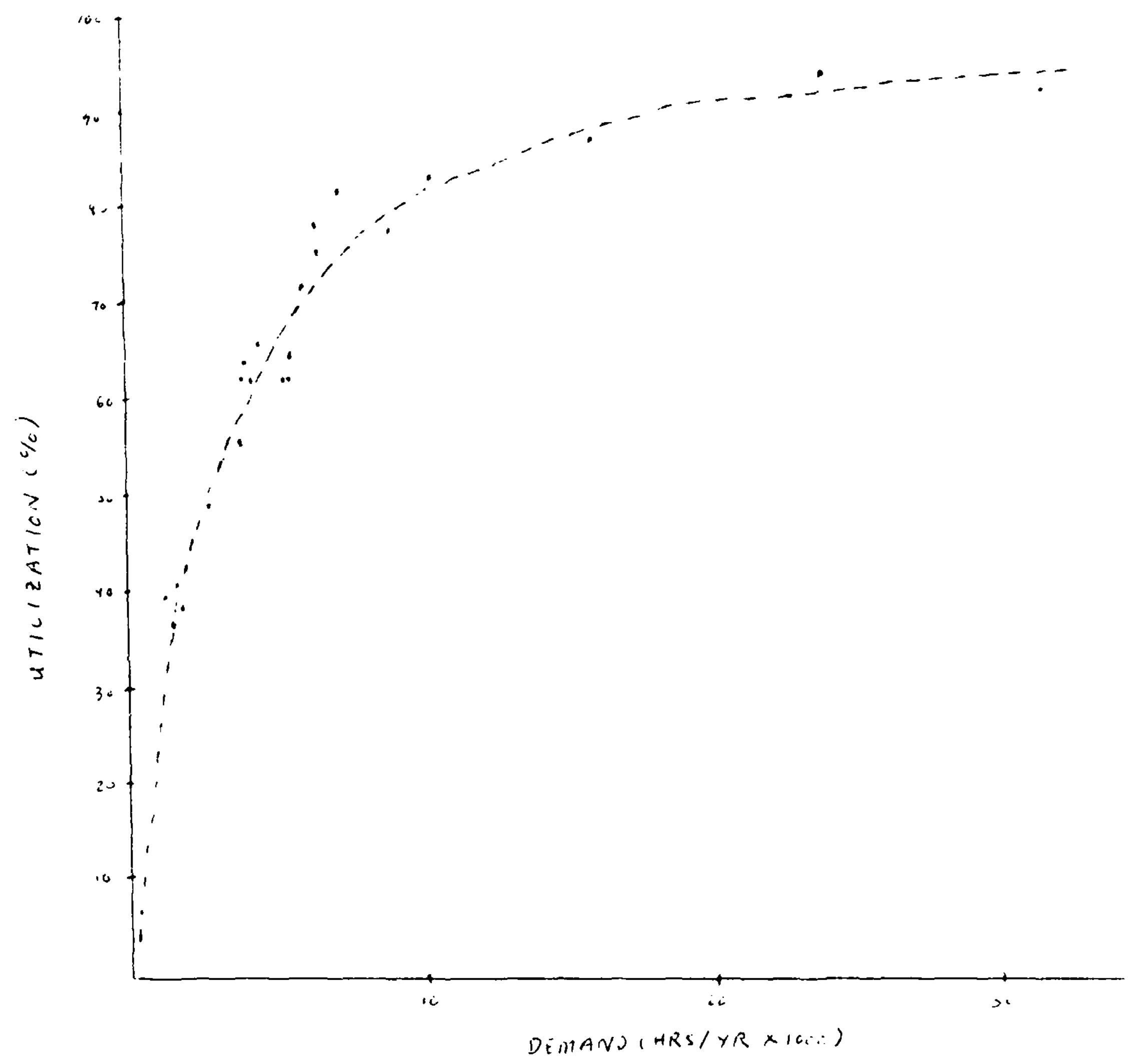


Figure 5.1 Utilization as a Function of Demand Load

Table 5.7 Required Inspectors and Utilization with Varied Demand

Demand Increment	Number of Inspectors	Utilization (%)
0.1	2	16.21
0.2	2	30.85
0.25	3	25.71
0.3	3	30.85
0.4	3	41.40
0.5	3	51.42
0.6	4	46.28
0.7	4	53.99
0.8	4	61.71
0.9	4	69.42
1.0	5	61.71
1.1	5	67.88
1.2	6	61.71
1.3	6	66.85
1.4	6	71.99
1.5	6	77.13
1.6	7	70.52
1.7	7	74.93
1.8	7	83.75
1.9	8	75.21

plateau is reached. For this data set, the plateau appears to be around 80%.

To gain full appreciation for the abrupt nature of the cycles, additional runs were made to show how local extremes can be determined. Using the data from Table 5.7, somewhere between 0.5 and 0.6 times the original demand requires the addition of a fourth inspector. As the demand load is slowly increased between these two amounts, a better approximation of the true maximum utilization is achieved. In this case, 0.58 results in three inspectors with 59.65% utilization and 0.59 still results in

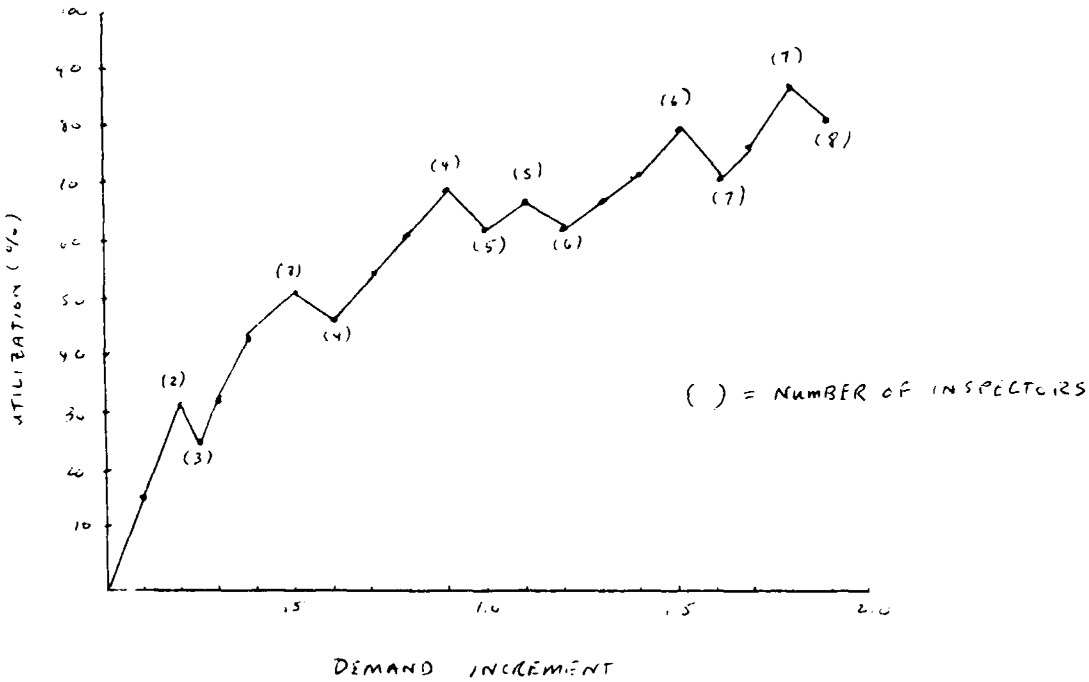


Figure 5.2 Utilization as a Function of Varied Demand

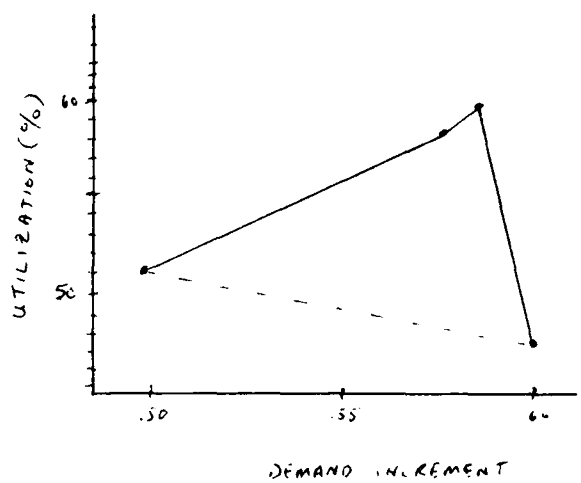


Figure 5.3 Maximal Utilization with Varied Demand

three inspectors increasing utilization to 60.68%. Thus, it is seen that the true local optimum amount of demand is between 0.59 and 0.60 of the original demand set.

These results are shown graphically in Figure 5.3 and highlight more clearly the true character of utilization as a function of demand. Using this approach, it is therefore possible to approximate an amount of demand which increases inspector utilization. This type of approach can be extremely useful when the number of resources is fixed and the nodes of a network have the capability for increased demand.

5.4 Tour Improvement

If the utilization curve as a function of demand load is to achieve the highest possible plateau, the tours assigned to inspectors must be optimal. As with any heuristic approach, the tours generated are not necessarily optimal. The savings approach used for tour generation in the computer program depends greatly on the order in which the nodes are assigned.

The tours generated for the twelve data sets presented above for determining the minimum number of inspectors were subjected to the 2-opt link exchange procedure of Lin [19]. Improvements in overall travel time were found to be negligible and certainly not of the magnitude to decrease the required number of inspectors to the lower bound estimates achieved by Dessouky et al.

However, to demonstrate the importance of tour improvement, a "pathological" example is presented. To evaluate the strength

of the tour generating method used in the implementation, the network depicted in Figure 4.1 is again used. The demands at every node of the first depot, nodes 1 to 13, are set to weekly requirements of 0.1 hours and the total days of inspector availability is set equal to the number of working days. In this way, the data is set to ensure that a single inspector will be assigned to travel to all of the nodes of the first depot.

This data set was then applied to the algorithm to evaluate the tour generation procedure in Chapter 3. A total of three tours were generated for a single inspector to meet the demands at each of the 13 nodes. The tours using the maximum savings approach were:

(1)	Node 1	Time spent:	0.0 hours
	Node 7	Time spent:	0.1 hours
	Node 6	Time spent:	0.0 hours
	Node 5	Time spent:	0.1 hours
	Node 6	Time spent:	0.1 hours
	Node 4	Time spent:	0.1 hours
	Node 3	Time spent:	0.1 hours
	Node 2	Time spent:	0.1 hours
	Node 1	Time spent:	0.0 hours
		Total travel time:	200 minutes
(2)	Node 1	Time spent:	0.0 hours
	Node 8	Time spent:	0.0 hours
	Node 9	Time spent:	0.0 hours
	Node 10	Time spent:	0.0 hours

Node 11	Time spent:	0.0 hours
Node 12	Time spent:	0.1 hours
Node 11	Time spent:	0.1 hours
Node 10	Time spent:	0.1 hours
Node 9	Time spent:	0.1 hours
Node 8	Time spent:	0.1 hours
Node 1	Time spent:	0.0 hours
	Total travel time:	180 minutes
(3) Node 1	Time spent:	0.0 hours
Node 13	Time spent:	0.1 hours
Node 1	Time spent:	0.0 hours
	Total travel time:	120 minutes

The overall travel time assigned to this inspector is 500 minutes. The tours divide the nodes of the first depot into three connected graphs. While the first two tours appear reasonable, merely by inspection of the second tour, it is clear that some improvement is probable.

To illustrate how the maximum savings method generates an inferior tour, we examine the construction of the second tour. By assigning the inspector to each node by their numerical order, node 8 was handled first. Since no savings was possible by including node 8 into the first tour, the second tour was spawned (1 - 8 - 1). Node 9 is assigned next. The maximum savings was found to occur if it was inserted into the link 1 - 8 yielding (1 - 8 - 9 - 8 - 1). Since node 8 must be traversed on the shortest path between node 1 and 9, no time is spent on the first visit to

node 8. Continuing in this manner, node 10 is inserted into the link 8 - 9 and traverses node 9 in its shortest path between 8 and 10 resulting in the tour (1 - 8 - 9 - 10 - 9 -8 - 1). Now no time is spent on the first visits to both nodes 8 and 9 but here we see that the direct link between nodes 1 and 10 is shorter than the path which has been generated. The same approach is followed for nodes 11 and 12 which generates the second tour (1 - 8 - 9 - 10 - 11 - 12 - 11 - 10 - 9 - 8 - 1) where no time is spent at any of the first four nodes and the alternative direct links are again ignored.

Using the systematic approach of Lin, we would like to explore in greater detail how this heuristic approach is applied to improve this tour along with the other two. Starting with the first tour we have (1 - 7 - 6 - 5 - 6 - 4 - 3 - 2 - 1). Successive exchanges of two branches at a time produce only tours of greater travel time so the procedure terminates with this tour as the best found. Indeed, this tour was optimal to start with so no improvement was possible.

For the second tour, the exchange procedure produces an optimal solution. Starting with the original tour of 180 minutes travel time, the best tour formed is another tour of the same length:

(8 - 9 - 10 - 11 - 12 - 11 - 10 - 9 - 1 - 8).

Since this is not the original tour, the 2-opt procedure is then applied to this tour. The best tour which results in another tour of the same length:

(9 - 9 - 10 - 11 - 12 - 11 - 10 - 9 - 1 - 8).

Here we consolidate the link 9 - 9 to a single visit to node 9 by combining the two times spent at node 9, shortening the tour by one node. Again, since this is not the original tour, we apply the procedure again which produces three tours of length 165 minutes. Branch exchanges were then applied to each of these three procedures which produced best tours of 145 minutes. Continuing the procedure at this point yielded subsequent repeats of the same tour, terminating the procedure.

The best tour generated by Lin's 2-opt procedure was the following tour of 145 minutes:

(11 - 12 - 10 - 9 - 8 - 1 - 11)

which translates to

(1 - 11 - 12 - 11 - 10 - 9 - 8 - 1)

in the standard notation with the depot as the originating point and all nodes of the shortest paths included.

The savings of 35 minutes in travel time is a 7% reduction. Translated to slack time over 252 total working days, the weekly savings (52 times per year) yields an additional 30.33 hours per year that could be spent by an inspector performing inspections. As seen from the example, even this amount of improvement can be important in the determination as to whether an inspector is assigned.

It is thus expected that over larger networks where ordering of nodes to ensure proper algorithm performance becomes quite impractical, the use of even a simple heuristic approach such as

the 2-opt exchange procedure can produce significant tour improvement. When tour improvements are transferred back to the load assignment procedure, inspector utilization of slack time is increased and the utilization curve achieves a higher plateau.

6. CONCLUSIONS AND RECOMMENDATIONS

This thesis presented a complex inspector resource allocation problem with multiple depots which, while similar in many respects to the generic vehicle routing problem, is not addressed directly in current literature. The major shortfalls of existing approaches in solving the problem formulated in Chapter 1 were due to the cyclic demand schedule combined with the overall objective of determining the minimum number of inspectors. Thus, the methodology used by Dessouky et al. [12] for the single depot inspector allocation problem provided the foundation for developing a solution method for the current problem with multiple depots.

In the process of extending their method to incorporate multiple depots, their method was expanded and modified to produce the algorithm which has been presented. This algorithm was then implemented on the computer to evaluate it over a wide variety of demand loadings. While overall performance was found to be inferior to the methodology used by Dessouky et al in determining the minimum number of inspectors, it did determine the optimal solution on half of the data sets provided. As with any heuristic approach, there is no guarantee of an optimal solution. However, great care has been taken to assure a feasible solution.

The parametric analysis which was conducted demonstrates the utility of the algorithm as a tool for making planning and management decisions. The program is used to determine the

number of resources and show how the number required is reduced with increased utilization. The effect of increased demand loads through the combining of separate skill categories clearly demonstrates how increased utilization is directly related to reducing the overall number of inspectors. When combined with the cost factors involved, the algorithm becomes a useful decision making tool. Further, by charting the behavior of utilization rates as a function of demand, a process is developed to determine demand levels which provide locally optimal utilization rates.

As seen by the parametric analysis, utilization reaches a plateau as demand is increased. The closer the tour assignment approach is to the optimal solution, the higher the plateau. As the basic procedure of the algorithm states, tour improvement should be made after each tour assignment and the net savings translated into additional inspector availability for the specific period being assigned. The implementation will thus be extended to incorporate tour improvement using the 2-opt exchange approach after each inspector is assigned as demonstrated in Chapter 5. In this way, the effect of node ordering on the quality of tours will be minimized.

The evaluation of this algorithm, for the most part, focused on data sets used by Dessouky et al to provide a basis for comparison. Even though the results were generally inferior, the method is believed to have potential. Therefore, it is recommended that further evaluation of the procedure be directed

in three broad directions:

(1) The networks used in this study were comparatively small for an overall evaluation of algorithm performance. The algorithm must be applied to much larger networks for a complete evaluation.

(2) The determination of the depots and the nodes assigned to the depots was assumed to be given for the problems presented. The algorithm could be extended to make one or both of these determinations prior to the assignment of inspectors.

(3) The modified savings method used for tour generation combined with the 2-opt exchange procedure for tour improvement were the only tour heuristics evaluated in this study. Other combinations of heuristics may be found to provide even better results.

In conclusion, this thesis presented an algorithm for solving the inspector resource allocation problem for multiple depots. The program which was developed yields "good" overall performance and has numerous applications in resource allocation planning. It is hoped that this study gives some definitive guidelines as a basis for further research in the broad class of problems known as the periodic vehicle routing problem.

APPENDIX A: PROGRAM DOCUMENTATION

DOCUMENTATION

The program ISTAR which implements the proposed algorithm is organized into eight separate program modules. Several of the modules include additional support subroutines which are shown below.

Program Modules and Supporting Subroutines

Module	Subroutine/Function	General Purpose
INSDAT	ADDINS	Adds inspector to provide slack coverage for total working period
	SORTIN	Sorts available inspectors according to total time available
ASSIGN1	SLKINSP	Finds an inspector with enough slack time for daily assignments or calls ADDINS to add another
	LOCDAT	Stores summary of assignments made
	SLKDAT	Summarizes of periods of constant slack time for each inspector
	LOOK	Adds additional inspectors as each new depot block is encountered if demand is greater than total slack
TOURS	SAVING	Function to calculate time savings
	FNDPTH	Determines shortest distance path
	SAVMAX	Determines maximum savings if new node is inserted into a given tour
	SRCHP	Determines if a node is already visited in a given tour
	CALCDAY	Function to calculate days required to meet demand when time is limited

I. INPUT DATA

The input of all given parameters is done in the main program segment ISTAR. From the formulation in Chapter 1, the following variables must be assigned values:

(1) m, the number of nodes in the network

(2) D, the number of depots in the network

(3) $\{i : i \in \{1,\ldots,D\}\}$, the depots in the network

(4) W, the number of working days in the time period

(5) P, the number of frequency classes

(6) f_p, the frequency of visits over the total time period required to perform class p inspections

(7) d_{ip}, the duration of class p inspection at node i

(8) h_{ip}, the need for a class p inspection at node i

(9) a_{kw}, the time available for inspector k on day w

(10) t_{ij}, the travel time between nodes i and j

The program variables which are used to represent each of these input parameters are given below.

	Input Parameter	Program Name
(1)	m	NN
(2)	D	NUMDEP
(3)	$\{i : i \in \{1,\ldots,D\}\}$	DEPLIST(I)
(4)	W	DR
(5)	P	P
(6)	A	TDA
(7)	f_p	FP(P)
(8)	d_{ip}	DEM(I,P)
(9)	h_{ip}	[Given by (8)]
(10)	a_{kw}	SLACK(K,W)
(11)	t_{ij}	TT(I,J) [Shortest path]

Initially, SLACK(K,W) is set to the maximum daily time

available as prescribed by the variable TAMAX. The variable TA carries the constant daily time available, SLACK(K,W), for a particular inspector over a specific period of days w. All times t_{ij} are input in minutes and are adjusted to hours in the program.

This implementation assumes the depots and the node assignments to depots have been made at the outset. The node indices for the demand and travel arrays must be ordered such that the depot for a block of nodes precedes the nodes to which it is linked. Until the next depot is indexed, no other depot's nodes are indexed. The ordering of nodes within the each depot block is arbitrary. In this way, DEPLIST(I) contains the indices for all the depots ordered sequentially. Thus, the nodes i of the depot block belonging to the depot denoted as DEPLIST(I) are

$$\text{DEPLIST(I)} \leq i < \text{DEPLIST(I+1)}.$$

For example, if nodes 5 and 12 are depots stored in DEPLIST(I) AND DEPLIST(I+1) respectively, then nodes 5 through 11 are the nodes local to the depot at node 5 which would be considered for demand satisfaction as a block. Node 12 is the next depot.

As shown above, the array TT(i,j) does not actually reflect the incidence of arc (i,j) but rather has the shortest path time between nodes i and j. Thus, even if a direct route does not exist between two nodes, the shortest path time can still be stored. This is accomplished by using any shortest path algorithm to determine both the optimal time and path between any two nodes of a network. With t_{ij} defined in this manner, a means

of storing the routing is also needed. A shortest chain matrix is the means by which the shortest path is obtained by this program. The information is stored in the array SC(i,j) such that SC(i,j) is the next intermediate node in the shortest path between nodes i and j.

II. DAILY ASSIGNMENTS

Every node within a depot block having positive daily demand is found in the DAILY module. For each node found, a call is made to the ASSIGN1 module where inspectors are assigned to satisfy all of the demand. The method used in ASSIGN1 is a load assignment procedure based on travel times and amount of demand which can be satisfied as determined by the TOURS module.

The model parameters used by the ASSIGN1 module and their program names are:

(1) The time available of inspector k on day w, a_{kw}, is represented by SLACK(K,W).

(2) The day subscript w is represented by the variable DAY.

(3) The inspector subscript k is designated by the variable NI.

(4) The daily demand at node i, d_{i1}, is denoted by the variable TDD for total daily demand.

(5) The variable n denoting the total number of inspectors is stored as TI.

(6) The total number of working days, W, is stored as DR, days required.

(7) The number of days an inspector is available for

inspections, A, is stored as TDA, total days available.

(8) The node i with demand to be met is represented by the variable INODE.

The summary of inspector slack is maintained in arrays which are generated by the subroutine SLKDAT. The array SLKSUM contains the constant value a_{kw} over a period of time. Using the same subscripts for all arrays, the inspector for which the period applies is stored in array SINSP, the first day of the period is contained in array SBEG and the last day in SEND. The total number of days is maintained in SDAYS. This summary method consolidates the data for convenient output and retrieval of inspectors with slack time. For example, if inspector 4 has 2.0 hours available per day from day 101 to day 200, SLKSUM(i) is 2.0, SINSP(i) is 4, SBEG(i) is 101, SEND(i) is 200, and SDAYS(i) is 100. The variable II is used as the common subscript for these arrays.

The assignment of inspectors to meet daily demand at a node INODE is accomplished in the ASSIGN1 module which implements the algorithm of Figure 3.2. ASSIGN1 first attempts to find an inspector from the slack summary whose first day of availability coincides with the first day required (initially set to day one). This determination and selection process is made by subroutine SLKINSP. If an inspector is not found, a new inspector is added whose first day of availability is the day required. The availability period then extends over the total availability period A or to the last working day, W, whichever occurs first.

The variable DA, the current number of days available, is set accordingly. The daily time available, TA, is then set to the initial daily availability TAMAX. If, on the other hand, an inspector is found with positive slack, a call to the TOURS module is made to ensure that travel time to INODE does not exceed the slack time by setting the TFLAG to 1. Provided there is sufficient slack, DA is set to the number of days in the slack period from SDAYS and TA obtains the daily slack time for the period as specified in SLKSUM.

Regardless of how an inspector with sufficient availability is obtained, subroutine SLKINSP then calls the TOURS module to make the assignment (TFLAG = 0) for the inspector. This call returns the travel times and amount of demand satisfied. The total daily demand is then reduced by the demand satisfied and the inspector's slack time for every day of the availability period is decremented by the total of demand satisfied and travel time to the node. The total travel time for an inspector k, t_k, is the sum of two travel variables, TRVL and ADTRVL. TRVL is set to the initial roundtrip travel time and ADTRVL is set to additional travel time if INODE was inserted into an existing tour. These two variables could have been represented as a single variable since only one will ever be greater than zero at a time.

To enforce the constraint that the number of days assigned for every inspector be less than or equal to A, the counter ICNT is used. While ICNT is less than the number of days available,

DA, the inspector's slack is decreased by demand and travel time until the last working day, W, is reached. If the total daily demand is not zero, more demand can be satisfied by the current inspector so another call is made to the TOURS module to make another tour assignment for the remaining number of days available. The amount of demand satisfied in this case is again subtracted from TDD and the process continues. If, at the last working day W, TDD is zero, all demand has been satisfied and a return to the DAILY module is made.

When the last day of the inspector's slack period is reached, indicated when ICNT is greater than the number of days available, DA, then another inspector must be used to complete the daily assignments if all required demand is not satisfied. Assuming more demand must be satisfied and the last working day, W, has not been reached, another inspector must be found with the additional constraint that the next inspector satisfy the same amount of demand as the previous inspector. This is accomplished by setting the global variable DFIRM to one which signals both the SLKINSP and TOURS subroutines that the demand value being sought is firm and not allowed to be reduced. In this case only, maximal use of existing tours may not be made to ensure constant demand satisfaction. If the last working day, W, has been reached and more demand must be satisfied, the TOURS subroutine is called with DFIRM set at zero and the entire procedure is repeated from day one. Finally, if the last working day is reached and all demand is satisfied, the subroutine returns to

the DAILY module.

The assignment of daily demand to an inspector is considered complete whenever A or W are reached by their respective counters. Each assignment is then recorded by the subroutine LOCDAT, detailing the location inspected, the inspector, the time period, the frequency period, the tour identification, and the number of days the inspection is conducted.

The nodes of a depot block with positive daily demand are retained for preference in the assignment of their non-daily demand. This record is maintained in the array of nodes visited, NV, subscripted by NNV, the number of nodes visited. The node indices of NV then become the first nodes to receive non-daily assignments.

III. NON-DAILY ASSIGNMENTS

The non-daily assignment procedure of Figure 3.3 is implemented by the ASSIGN2 module with help from the INSDAT program module. The first requirement for making non-daily assignments using the slack time of inspectors is to find a set of inspectors with continuous availability over all working days. This requirement is accomplished by the first portion of the INSDAT module. The arrays AVLINSP, DAYSAVL, and HRSAVL, indexed by the common subscript INSPCNT, are used to perform this function. Different combinations of inspectors with positive slack times, who provide continuous coverage, are surveyed from the slack summary arrays until the number of days of coverage is greater than or equal to the total working period. If total

coverage is not obtainable from the slack arrays, subroutine ADDINS is called to add an inspector with availability beginning on the day after the availability of the last inspector added ended (or day one for the first inspector added). AVLINSP receives the inspector number, DAYSAVL the number of days available in the slack period from SDAYS, and HRSAVL the slack time per day from SLKSUM.

Once a set of inspectors with continuous coverage is found, the number of inspections by each inspector that can be accomplished in a frequency period is calculated. The following program variables are used:

(1) TOTUSED(k,p) represents the parameter v_{kp}, the number of visits provided by an inspector k in period p

(2) DAYSAVL(k) represents d_k, the number of days inspector k is available in a particular constant slack period

(3) FP(p) represents f_p, the number of visits required in a period p

(4) HRSAVL(k) represents the formulation variable a_{kw}, the daily time available of inspector k where the days w range over the current availability period being considered.

(5) REMAIN(p) represents r_p, the number of visits remaining to be assigned in a period p. This array is initialized to the value of array FP every time a new combination of inspectors is considered. These arrays are then used to determine the number of visits an inspector k may perform in period p as presented in Chapter 3.

Once a set of inspectors is obtained with joint coverage of all working days, the ASSIGN2 module is called by the DAILY module for every node visited daily with positive non-daily demand and by the CYCLIC module for the remaining nodes of a block with non-daily demands. The first objective of the ASSIGN2 module is to determine if an inspector in the set visits the node requiring non-daily inspections in a higher frequency period. With TFLAG set at 2, calls to the TOURS module are made for each inspector to make this determination and assign travel time if not visited in a higher period.

The arrays DTRVL, DADTRVL, DSAT, and DTOUR are used in making the demand determination. These arrays are subscripted identically to AVLINSP using INSPCNT. DTRVL and DADTRVL receive travel time of TRVL and ADTRVL respectively, their sum again representing t_k. DSAT will receive the amount of demand an inspector can satisfy, D_{sk}. DTOUR receives the tour number of the inspector or zero if no tour exists.

In much the same manner as ASSIGN1, ASSIGN2 ensures that the demand is assigned evenly over the total working period. The next phase determines the amount of demand that can be satisfied by each of these inspectors. This minimum value is then assigned to all of the inspectors. When the amount of demand an inspector can satisfy is zero, the inspector lacks sufficient slack to travel to the node and perform inspections so a call is made to INSDAT to provide another set of inspectors.

The calculation of the amount of demand that can be

satisfied follows the procedure outlined in Chapters 3 and 4. The program variables which are used in making this determination are defined as follows:

(1) TOTSLK = $(t_k + d_{ip})v_{kp}$, the total time required to fulfill the demand requirement at node i in period p, incurring travel time of t_k

(2) TRVSLK = $t_k * v_{kp}$, the total travel time requirement

(3) SLKAVL = S_k, the total amount of slack time available for inspector k in the current availability period.

The amount of demand an inspector can satisfy is stored in the array DSAT. If an inspector's travel slack is greater than the total slack available, then DSAT is set to zero. If the total slack required is greater than the inspector's slack available, then the demand that may be met is limited by total hours available in the period. Otherwise, if the demand required is less than the total slack available and slack available is greater than the travel slack, all of the demand may be met. The minimum amount of demand satisfied of all the inspectors is the amount of demand that is assigned for the current iteration.

The final phase of ASSIGN2 makes the actual assignment of inspectors and adjusts their slack times. Only inspectors visiting a node for the first time, symbolized by their value of DTOUR being zero, are assigned tours by setting TFLAG to zero. The assignment data is finally recorded for each inspector of the set.

The amount of daily availability used by the assignment is

updated by the slack used per day. If D_s is the amount of demand satisfied in a period p, and t_k is the travel time required by an inspector, then the slack used per day by inspector k is

$$SLKPD = (d_s + T_k) * v_{kj} / D_k.$$

This amount of slack time is decremented from each day of the inspectors' availability periods.

When all of the inspectors with total coverage have been assigned, the total demand for the node is decreased by the amount satisfied. The procedure is repeated until all demand has been satisfied.

IV. TOUR GENERATION

The TOURS module is called by both the ASSIGN1 or ASSIGN2 module to satisfy demand for a given node. This module implements the algorithm depicted in Figure 3.4. When a call is made to this module, a specific inspector is considered who has some amount of slack time per day over a range of days. The TOURS module then makes a tour assignment using a maximum savings approach outlined in Chapter 3.

The TOURS subroutine is passed six arguments to accomplish this task. The inspector with slack is identified by number represented as NINSP in the subroutine. The frequency period for the demand node is variable J. The amount of demand which is to be satisfied is TUSED. The number of days that the inspector is to travel on the proposed tour is IDAYS. The inspector's availability period is begins at IDYBEG and ends at IDYEND.

In addition, the global variable INODE carries the index of

the node with demand currently being satisfied. The variable INNDTR is used to identify the tour number of the inspector to travel to the node for recording of the assignment at the higher level. When INNDTR is set equal to -1, it indicates that the inspector's own depot has been assigned and no tour assignment is made. Finally, the variable for an inspector's current daily time available is represented by TA in the program. This parameter provides the fundamental constraint for determining the maximum allowable length of a tour and the amount of time an inspector can spend traveling and inspecting.

The specific tour data is represented in the program as follows:

(1) The number of tours which exist for inspector k is maintained in array NTOUR as NTOUR(k).

(2) The number of nodes in tour t for inspector k is stored in array NNT as NNT(k,t).

(3) The nth node of tour t for inspector k is stored in array TOUR as TOUR(t,n,k).

(4) The tour frequency of tour t for inspector k, the number of days that the tour is available for use, is kept in array TRFRQ as TRFRQ(i,k).

(5) The first day tour t is available for use by inspector k is stored in array TRBEG as TRBEG(k,t).

(6) The last day tour t is available for use by inspector k is stored in array TREND as TREND(i,k).

(7) The time spent at the nth node of tour t for inspector

k is stored in array TSN as TSN(t,n,k).

(8) The maximum allowable time that inspector k can spend on tour t is stored in array TRLIM as TRLIM(k,t).

There are two possible purposes for making a call to TOURS. First, if an inspector is in consideration for an assignment to a node, the amount of slack time is checked first to be sure that it is greater than the time required to travel to the node. In this case, TOURS follows the procedure to minimize travel time but no assignment is made. This is accomplished through the use of a tour flag, TFLAG, which is set to a value of one or two in ASSIGN1 or ASSIGN2 respectively. The second purpose is to actually make a tour assignment for an inspector. In this case, TFLAG is set to zero by the higher levels.

Travel times are distinguished by the round-trip shortest path time between the inspector's depot, DEPOT(NINSP) and the node, INODE, versus the time for additional travel if the node, INODE, is incorporated into an already existing tour. The round-trip travel time is represented as TRVL. The additional travel time when incorporating into a tour is ADTRVL.

Whether called to check travel times or make an actual tour assignment, the heuristic optimization procedure is the same. When an inspector is to be assigned a tour, all of the inspector's tours which fall within the range of the current slack period are examined to determine: (1) if the node is already visited in an existing tour and (2) where the maximum savings can be achieved if the node is inserted into an existing

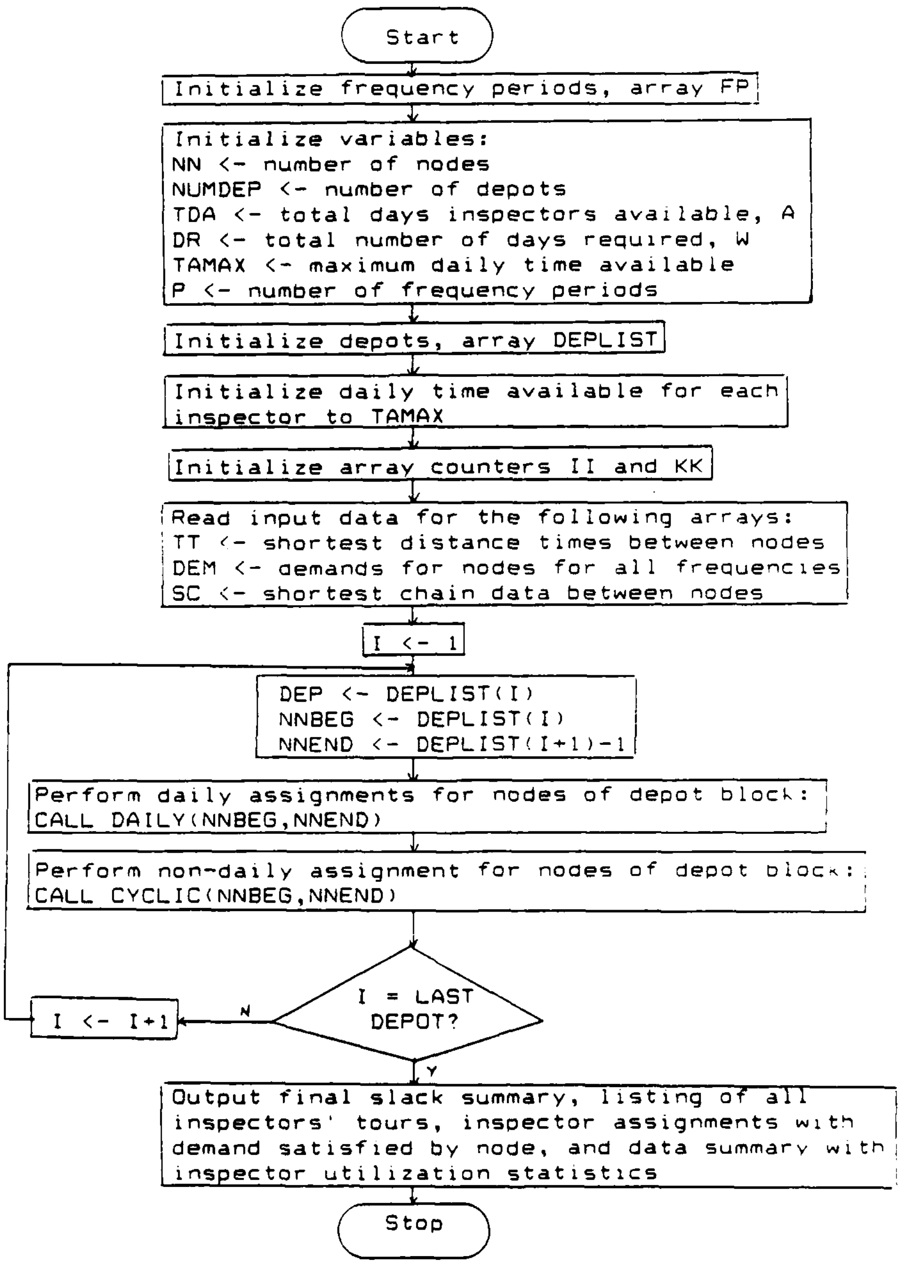
tour.

Roundtrip tours are originated when no savings can be achieved, the node is not already visited, or if no tour currently exists for the time period being considered. The subroutine FNDPTH is used to determine the shortest routing between any two nodes in the network using array SC. It is also used when a node is inserted into an existing tour.

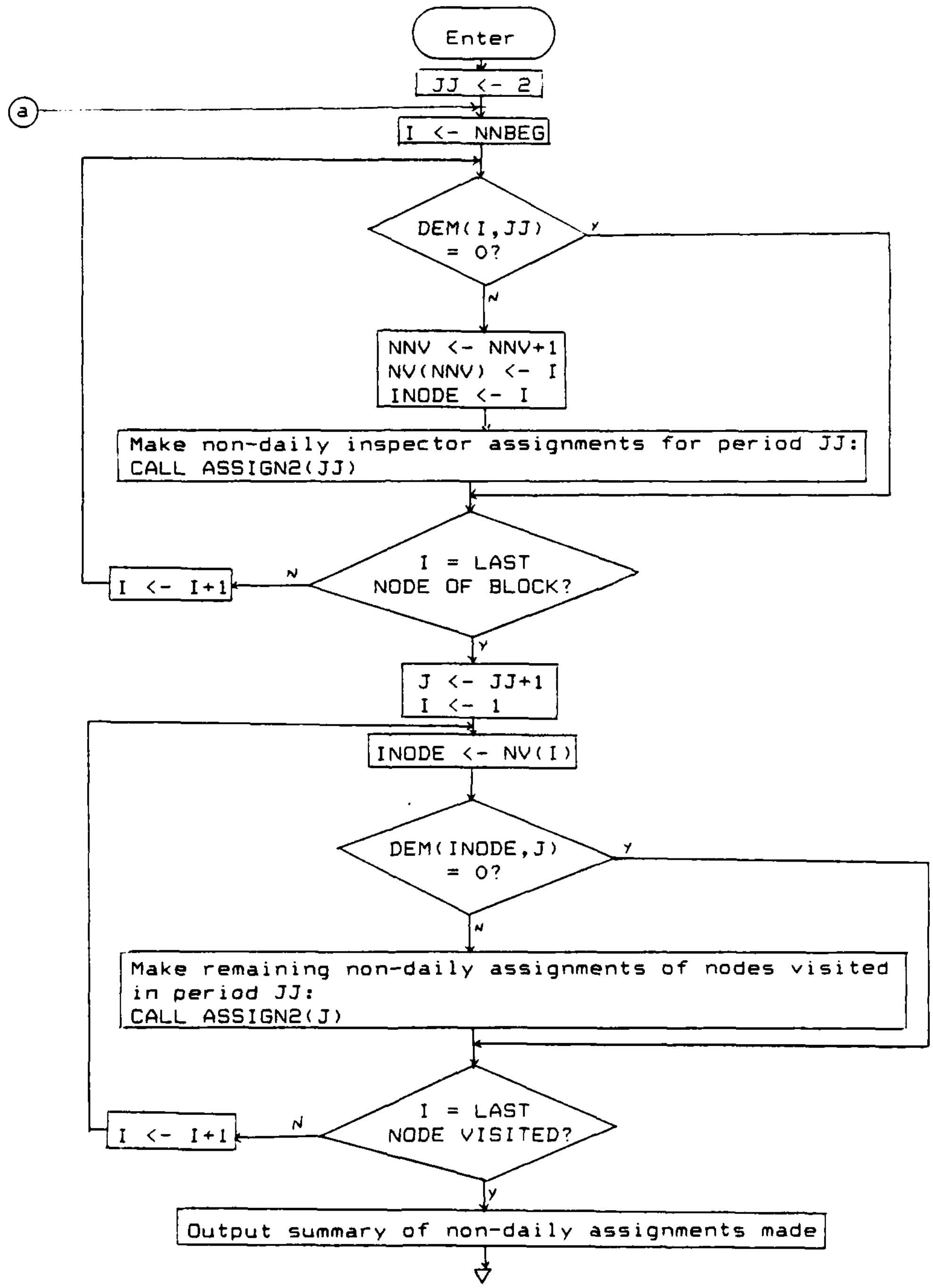
Using shortest path data instead of actual arc links, it is possible to visit a node before a demand assignment has been made. Subroutine SEARCHP is used to determine if a node is already included in an existing tour which falls within the range of availability. When the node is not already transited with time available on the tour, the same tours are explored to determine where the maximum savings can be achieved. This is accomplished by subroutine SAVMAX. Through repeated calls to the function SAVING to determine the savings incurred if the new node INODE is included between every pair of nodes of a tour, SAVMAX finds where the node can be inserted to achieve maximum savings, as defined in Chapter 3. By considering every feasible tour in this manner, the maximum savings over all tours is determined.

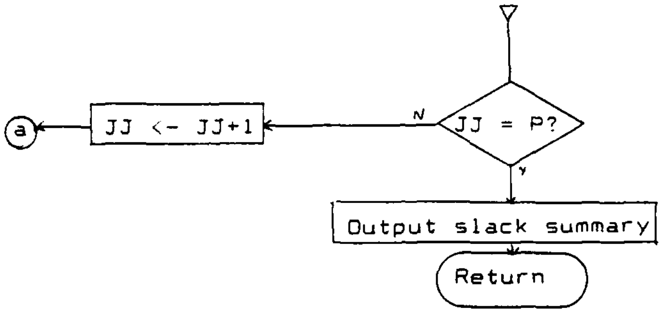
In the case of daily assignments only, TUSED, the amount of demand that is requested to be satisfied, may be reduced to meet an inspector's daily availability constraint. The amount of demand that may be satisfied for non-daily assignments is determined by the ASSIGN2 module.

MAIN PROGRAM MODULE ISTAR:

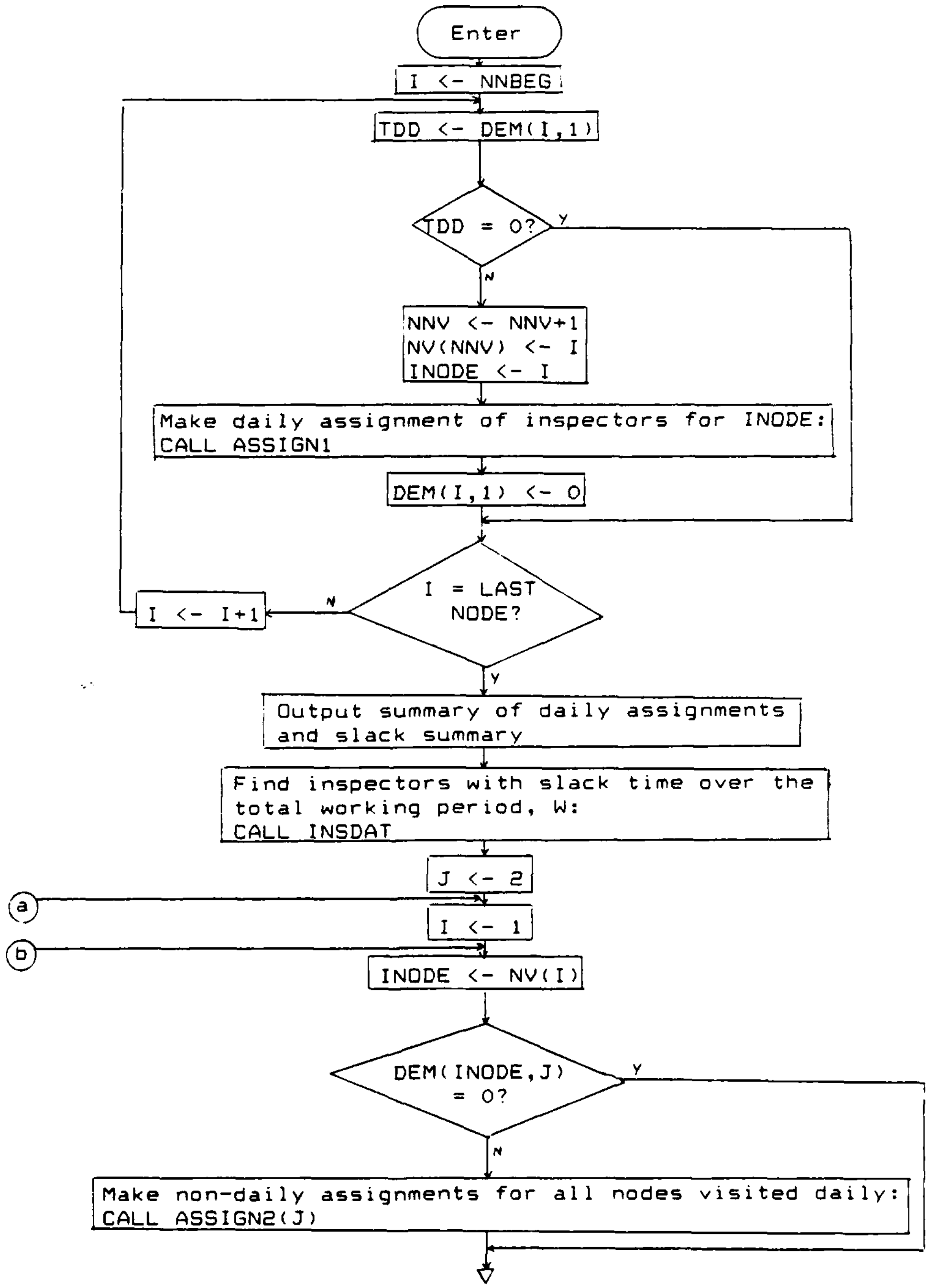


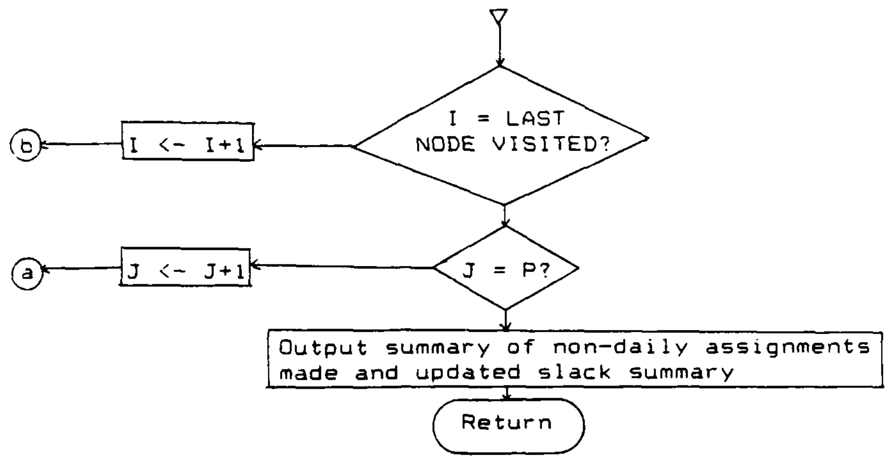
PROGRAM MODULE CYCLIC(NNBEG,NNEND):



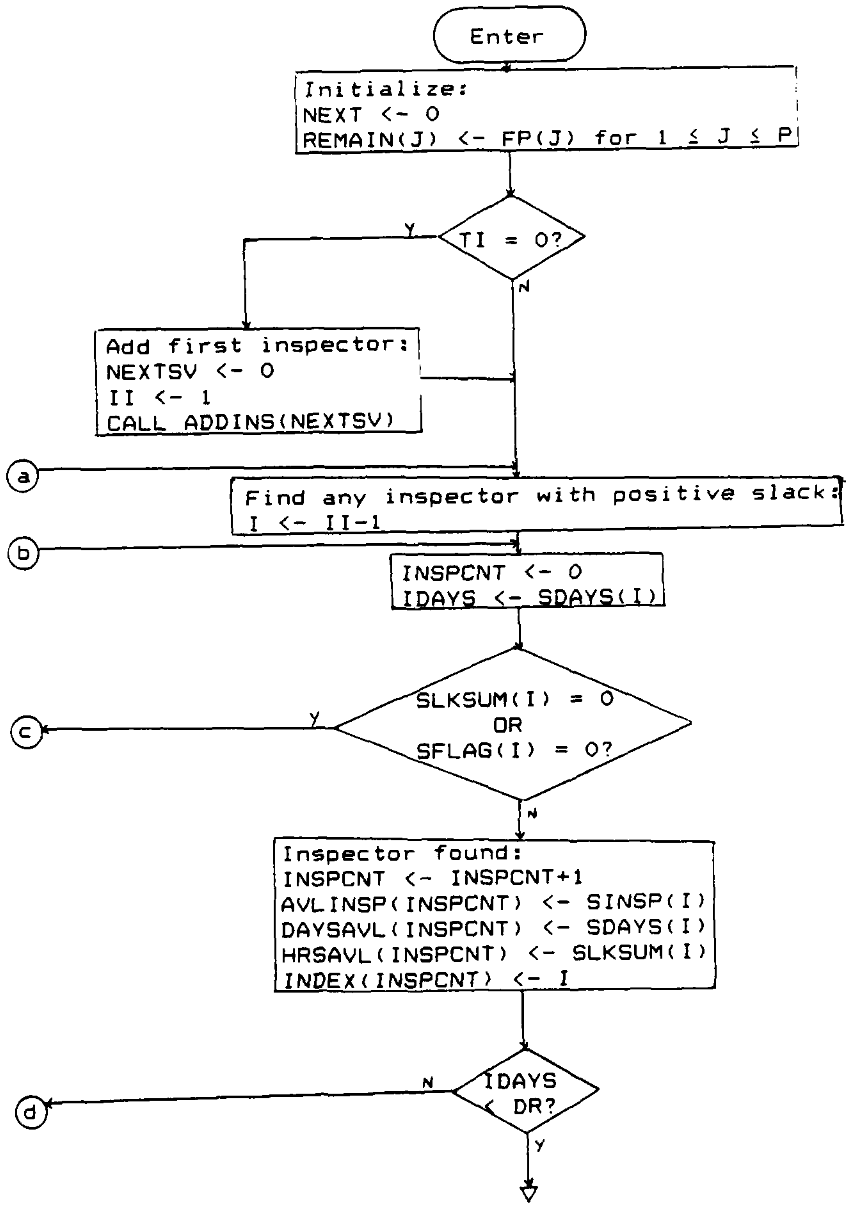

JJ = P?
N
Y
JJ <- JJ+1
a
Output slack summary
Return

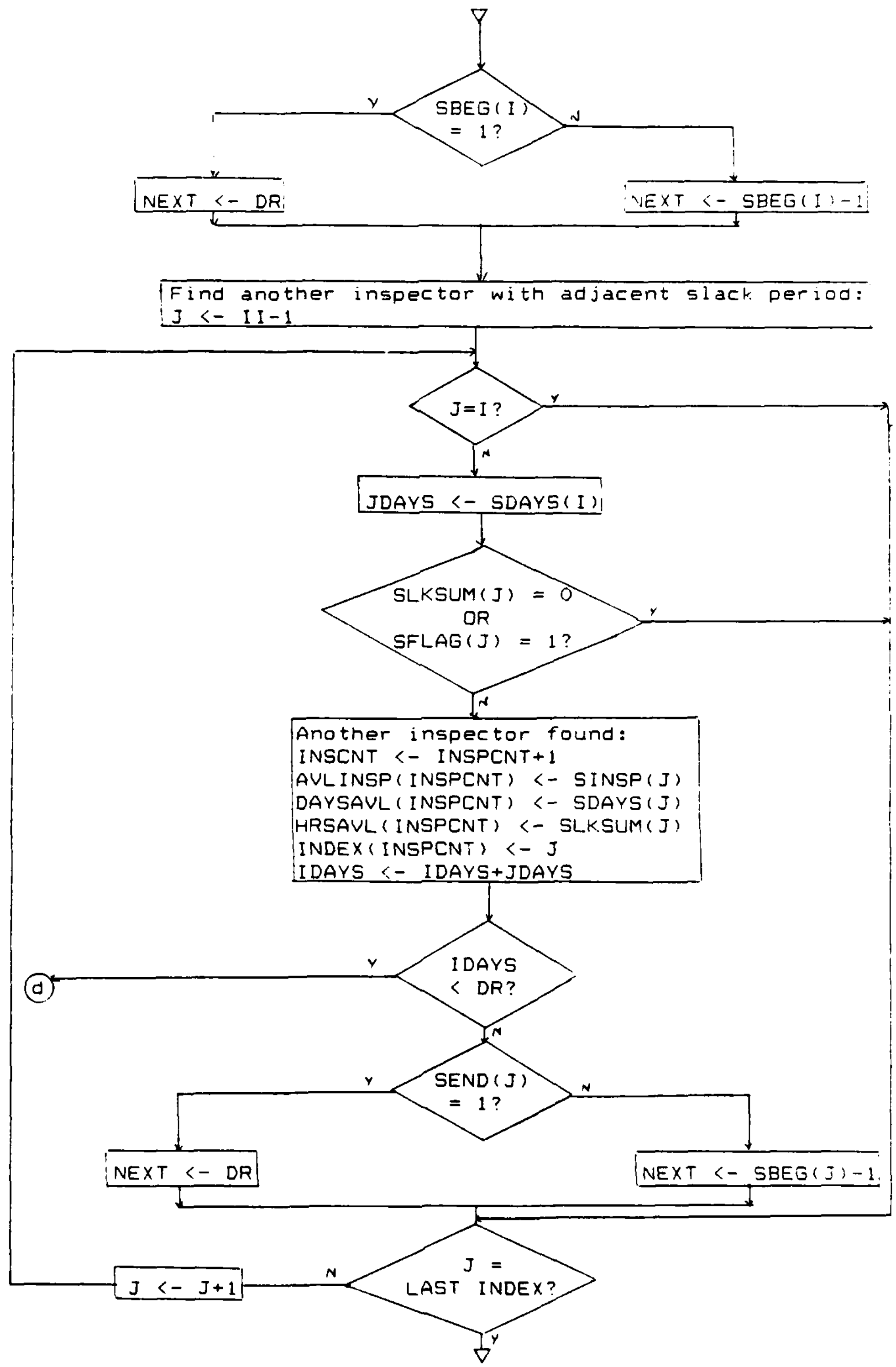
PROGRAM MODULE DAILY(NNBEG,NNEND):

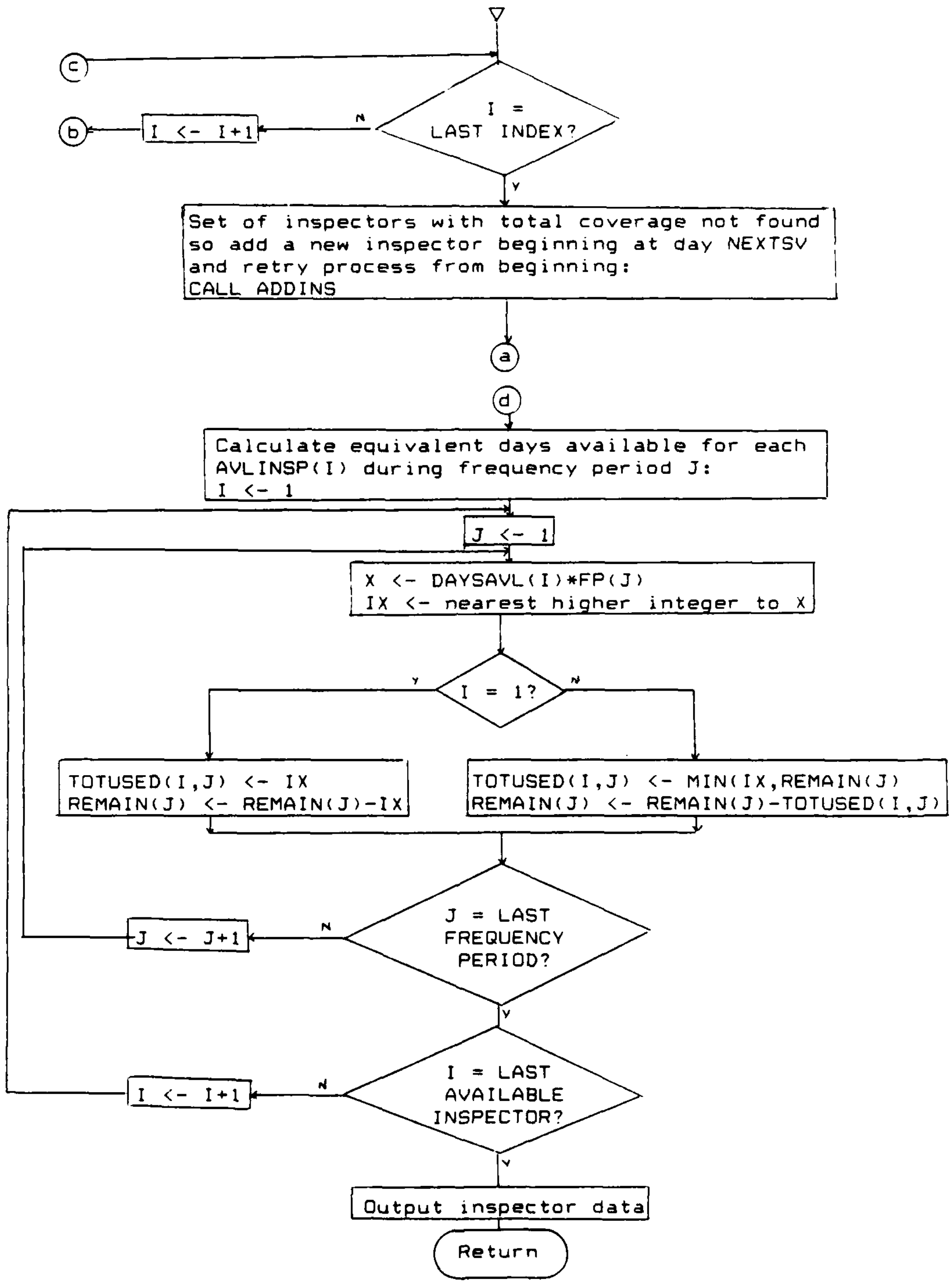



I = LAST
NODE VISITED?
b
I <- I+1
a
J <- J+1
J = P?
Output summary of non-daily assignments
made and updated slack summary
Return

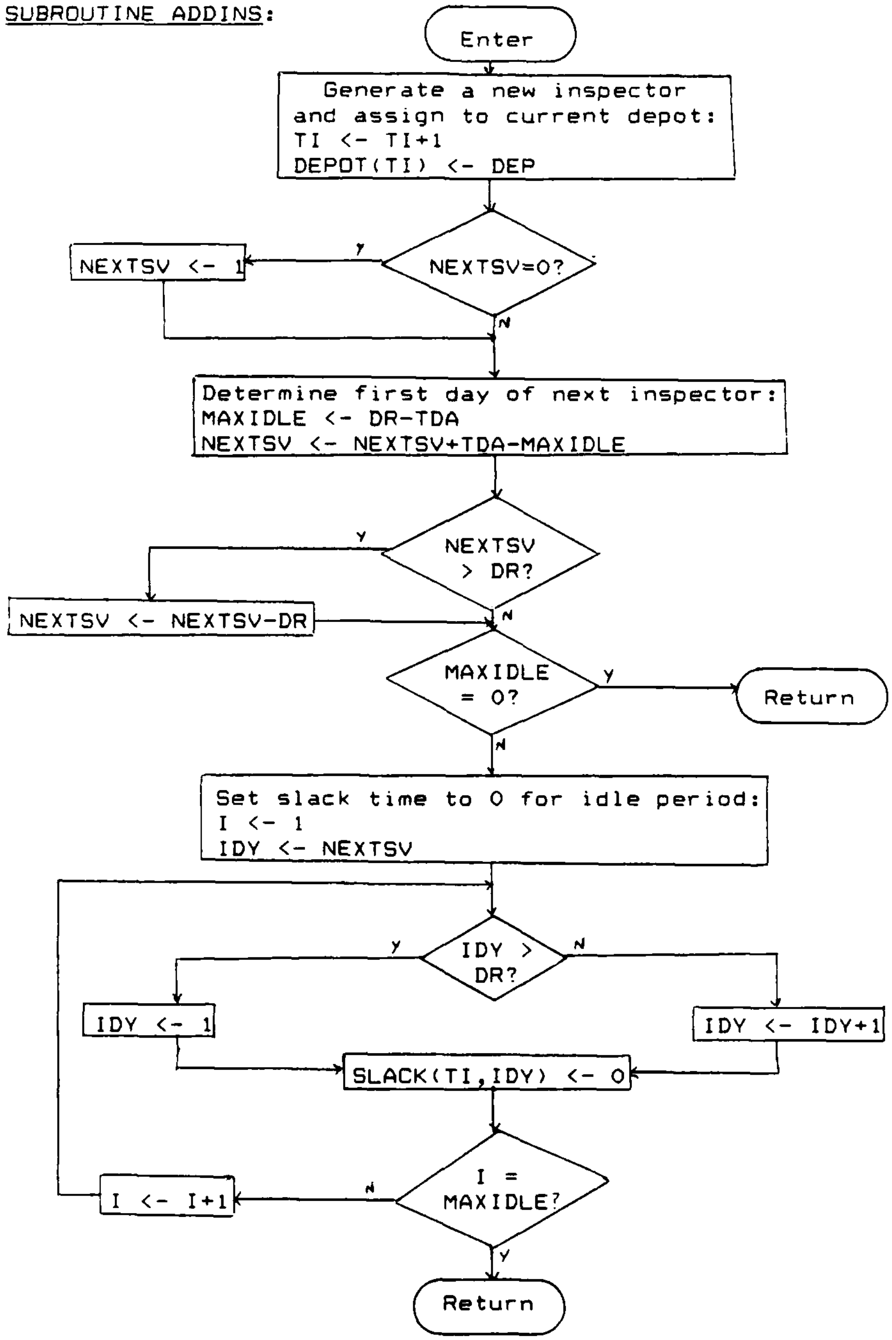
PROGRAM MODULE INSDAT:



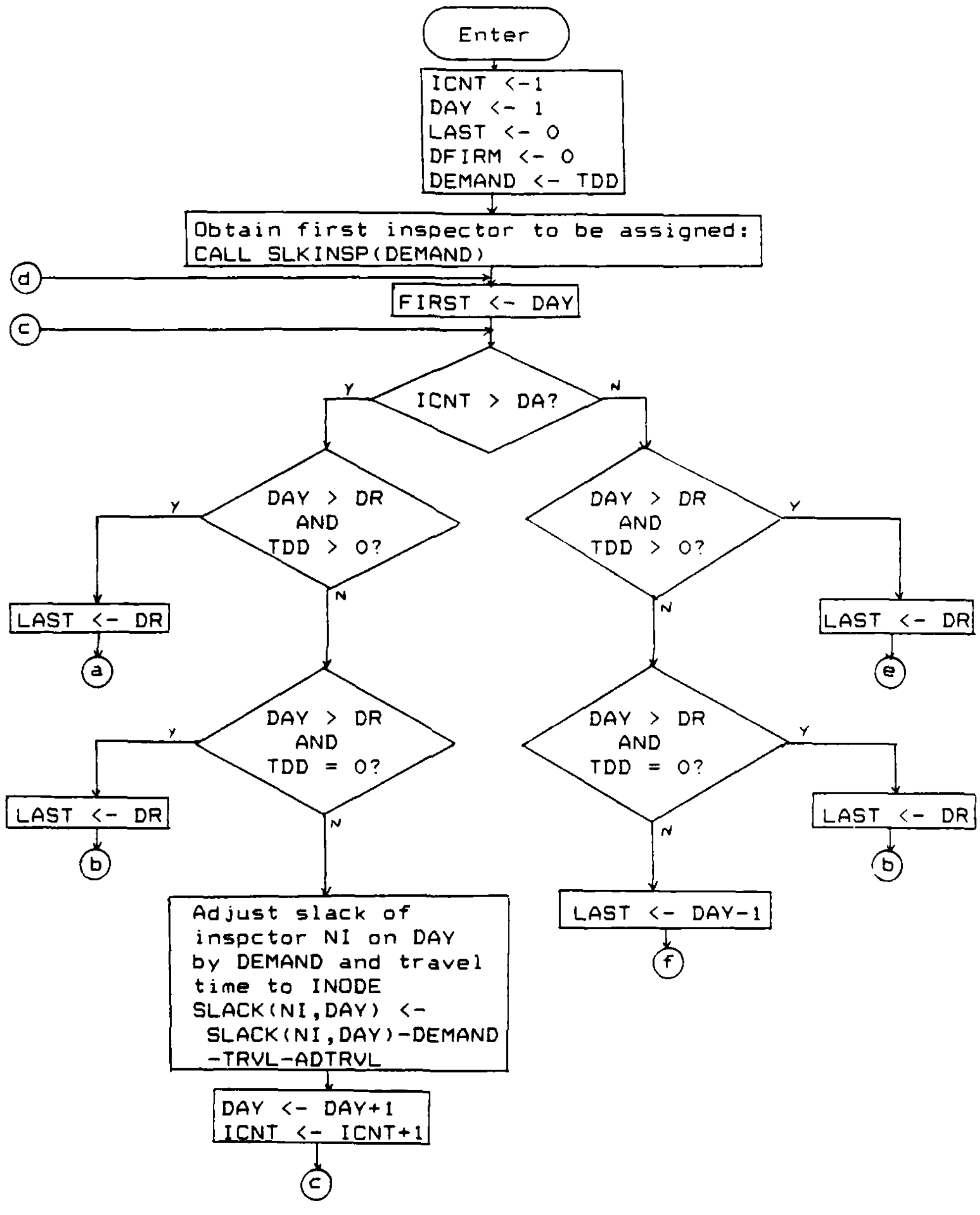

SBEG(I) = 1?
Y
N
NEXT <- DR
NEXT <- SBEG(I)-1
Find another inspector with adjacent slack period:
J <- II-1
J=I?
Y
N
JDAYS <- SDAYS(I)
SLKSUM(J) = 0
OR
SFLAG(J) = 1?
Y
N
Another inspector found:
INSCNT <- INSPCNT+1
AVLINSP(INSPCNT) <- SINSP(J)
DAYSAVL(INSPCNT) <- SDAYS(J)
HRSAVL(INSPCNT) <- SLKSUM(J)
INDEX(INSPCNT) <- J
IDAYS <- IDAYS+JDAYS
IDAYS < DR?
Y
d
N
SEND(J) = 1?
Y
N
NEXT <- DR
NEXT <- SBEG(J)-1
J = LAST INDEX?
N
J <- J+1
Y

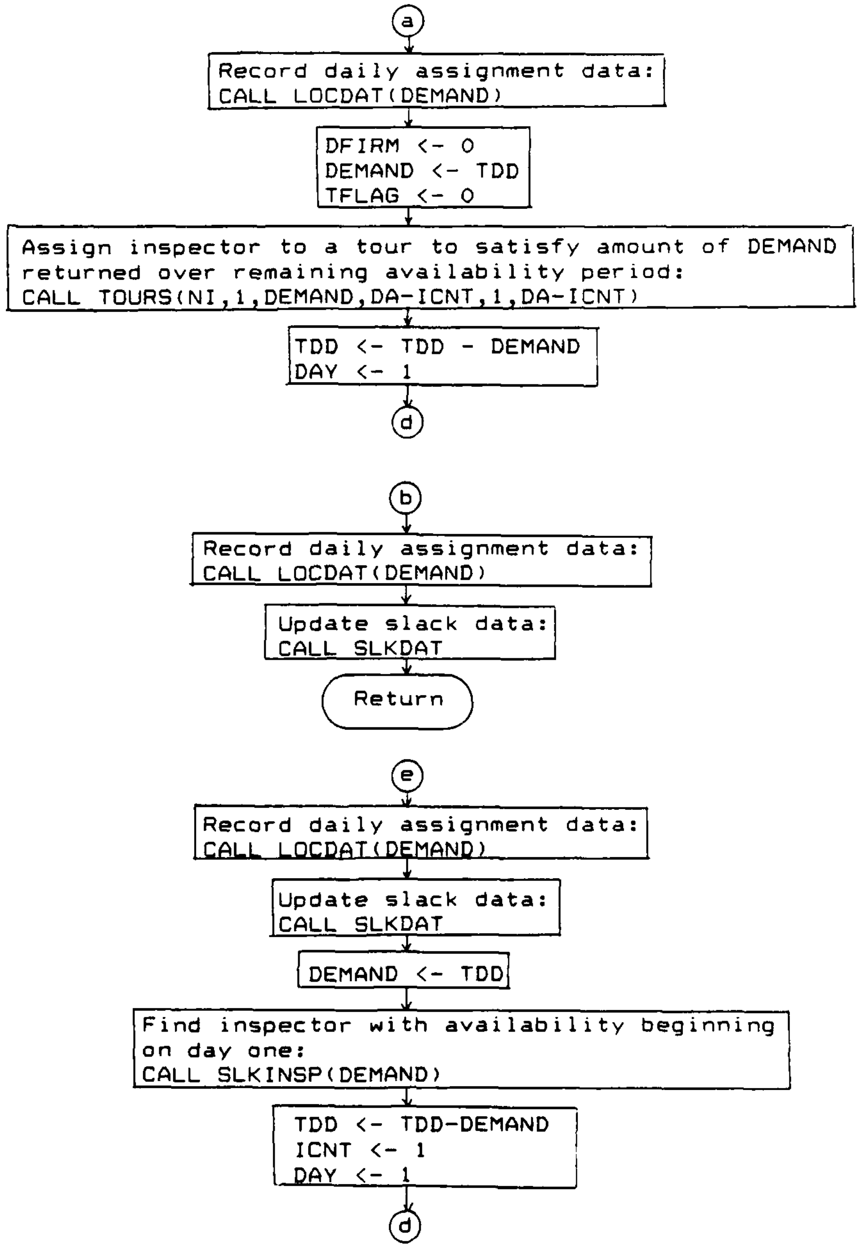

c
b
I <- I+1
N
I = LAST INDEX?
Y
Set of inspectors with total coverage not found so add a new inspector beginning at day NEXTSV and retry process from beginning:
CALL ADDINS
a
d
Calculate equivalent days available for each AVLINSP(I) during frequency period J:
I <- 1
J <- 1
X <- DAYSAVL(I)*FP(J)
IX <- nearest higher integer to X
Y
I = 1?
N
TOTUSED(I,J) <- IX
REMAIN(J) <- REMAIN(J)-IX
TOTUSED(I,J) <- MIN(IX,REMAIN(J))
REMAIN(J) <- REMAIN(J)-TOTUSED(I,J)
J <- J+1
N
J = LAST FREQUENCY PERIOD?
Y
I <- I+1
N
I = LAST AVAILABLE INSPECTOR?
Y
Output inspector data
Return

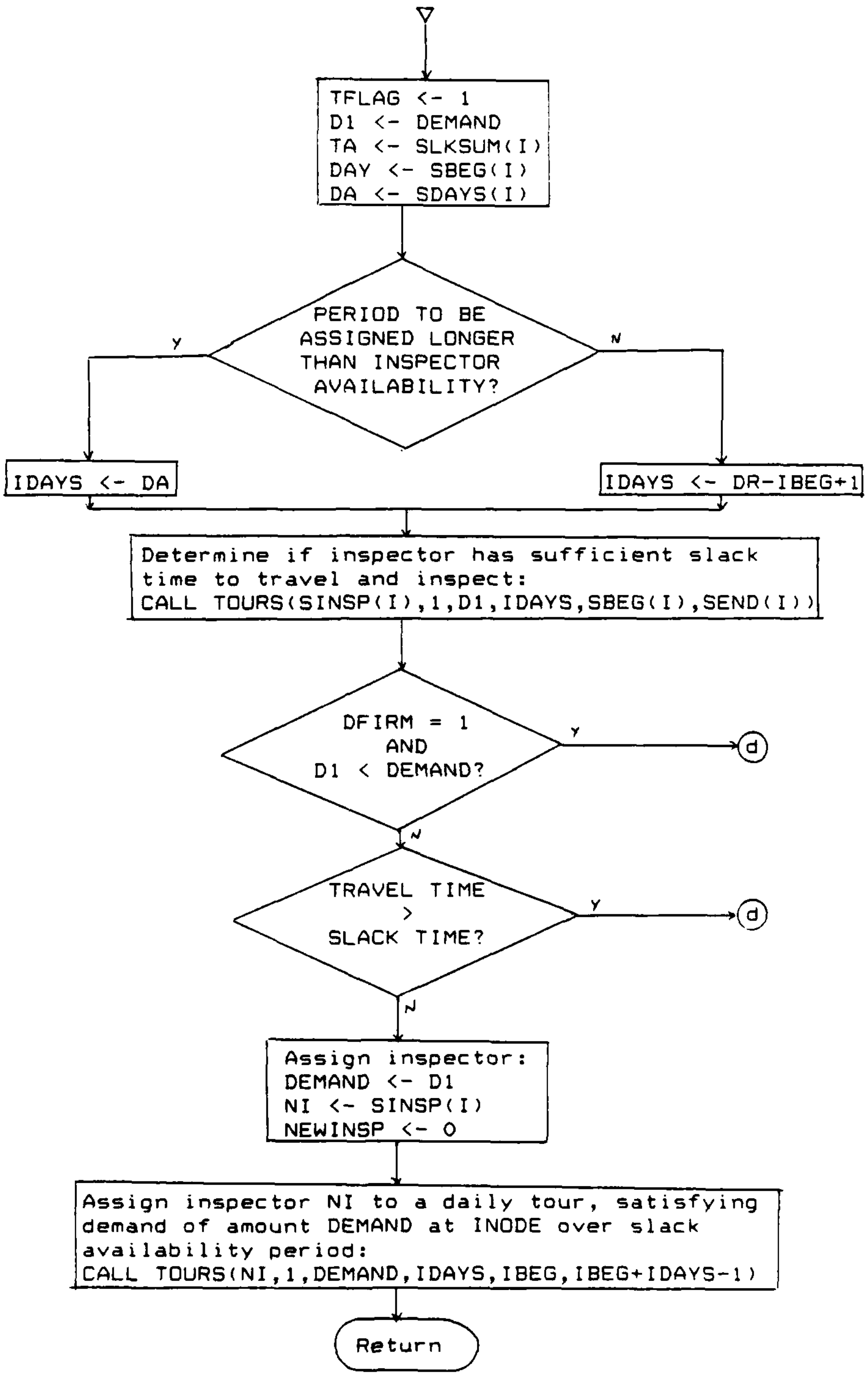
SUBROUTINE ADDINS:

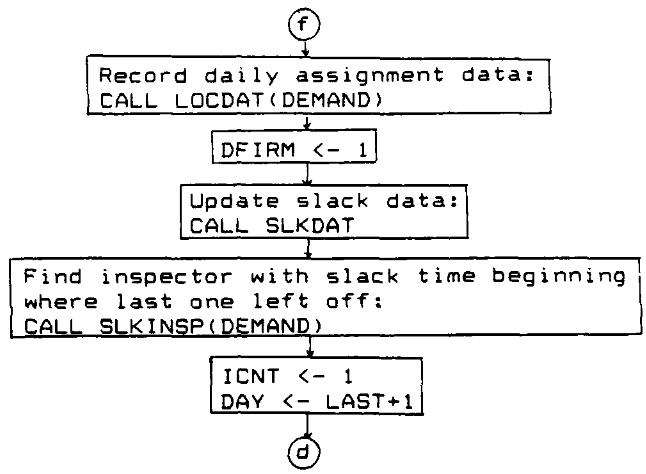


PROGRAM MODULE ASSIGN1:

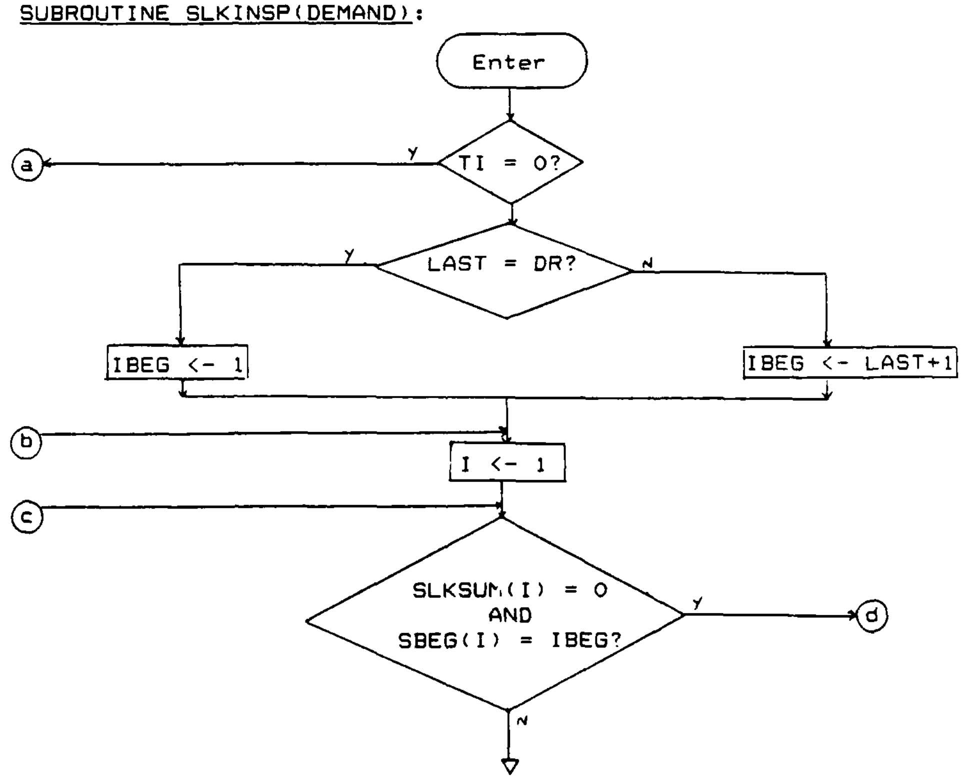


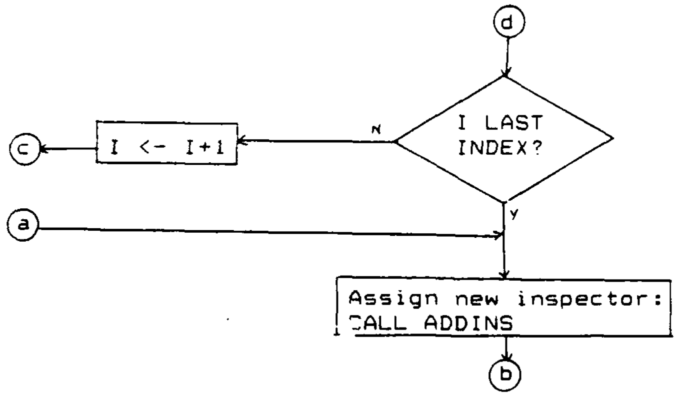

a
Record daily assignment data:
CALL LOCDAT(DEMAND)
DFIRM <- 0
DEMAND <- TDD
TFLAG <- 0
Assign inspector to a tour to satisfy amount of DEMAND
returned over remaining availability period:
CALL TOURS(NI,1,DEMAND,DA-ICNT,1,DA-ICNT)
TDD <- TDD - DEMAND
DAY <- 1
d
b
Record daily assignment data:
CALL LOCDAT(DEMAND)
Update slack data:
CALL SLKDAT
Return
e
Record daily assignment data:
CALL LOCDAT(DEMAND)
Update slack data:
CALL SLKDAT
DEMAND <- TDD
Find inspector with availability beginning
on day one:
CALL SLKINSP(DEMAND)
TDD <- TDD-DEMAND
ICNT <- 1
DAY <- 1
d


TFLAG <- 1
D1 <- DEMAND
TA <- SLKSUM(I)
DAY <- SBEG(I)
DA <- SDAYS(I)
PERIOD TO BE ASSIGNED LONGER THAN INSPECTOR AVAILABILITY?
Y
N
IDAYS <- DA
IDAYS <- DR-IBEG+1
Determine if inspector has sufficient slack time to travel and inspect:
CALL TOURS(SINSP(I),1,D1,IDAYS,SBEG(I),SEND(I))
DFIRM = 1 AND D1 < DEMAND?
Y
d
N
TRAVEL TIME > SLACK TIME?
Y
d
N
Assign inspector:
DEMAND <- D1
NI <- SINSP(I)
NEWINSP <- 0
Assign inspector NI to a daily tour, satisfying demand of amount DEMAND at INODE over slack availability period:
CALL TOURS(NI,1,DEMAND,IDAYS,IBEG,IBEG+IDAYS-1)
Return

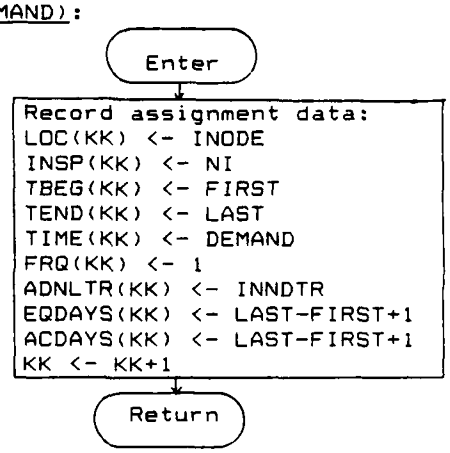


SUBROUTINE SLKINSP(DEMAND):

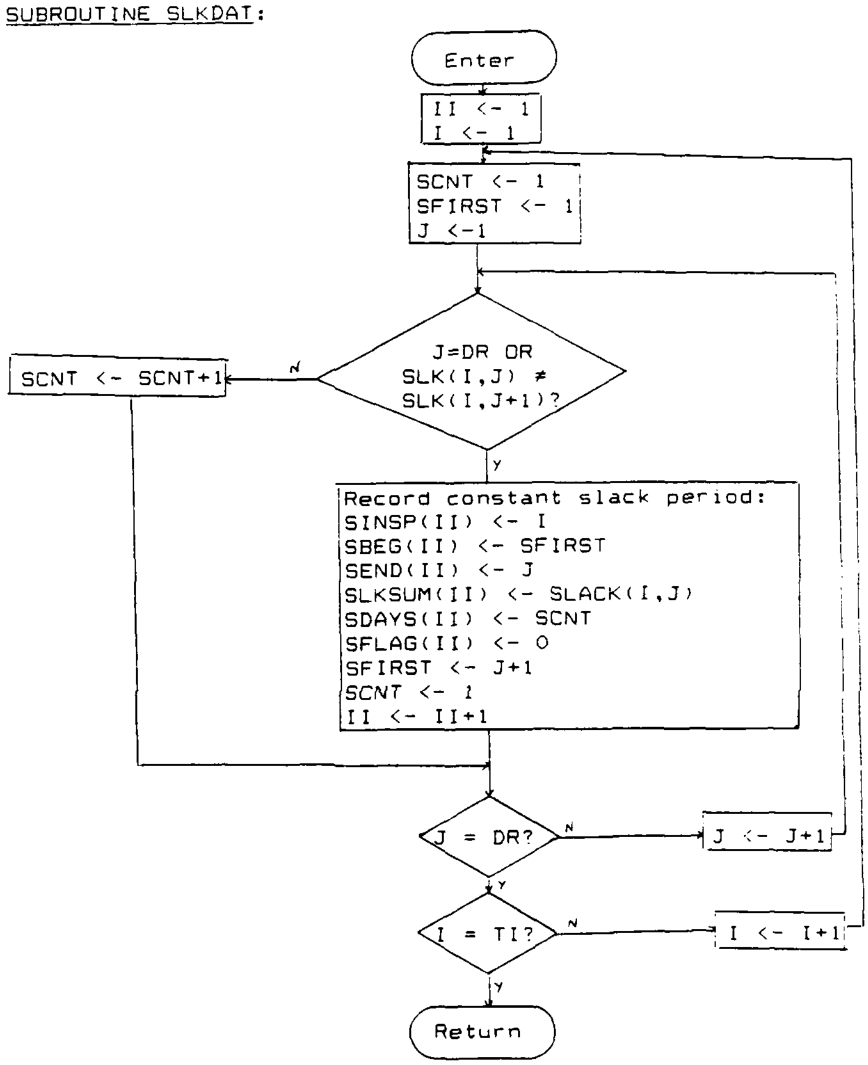




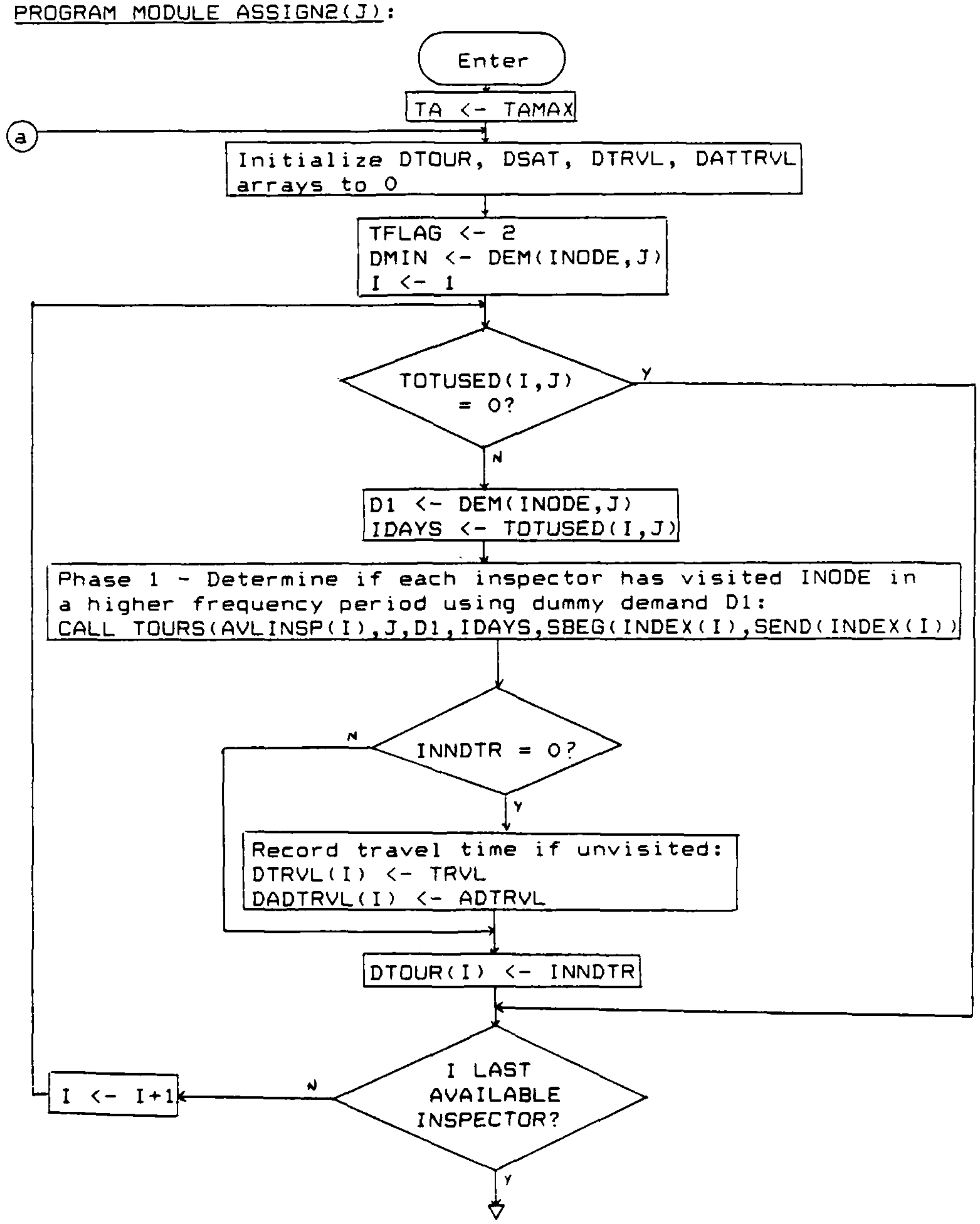
SUBROUTINE LOCDAT(DEMAND):

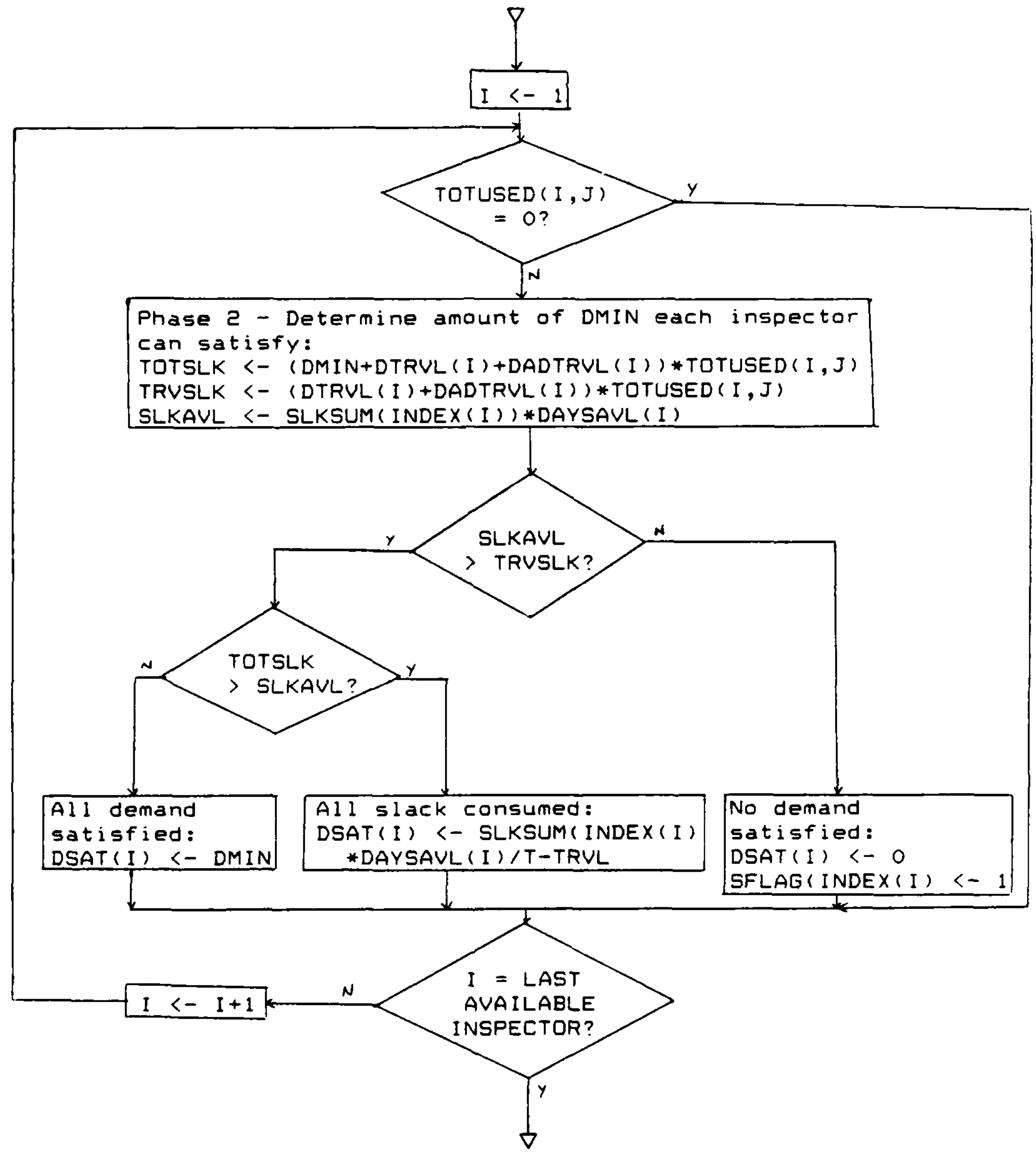


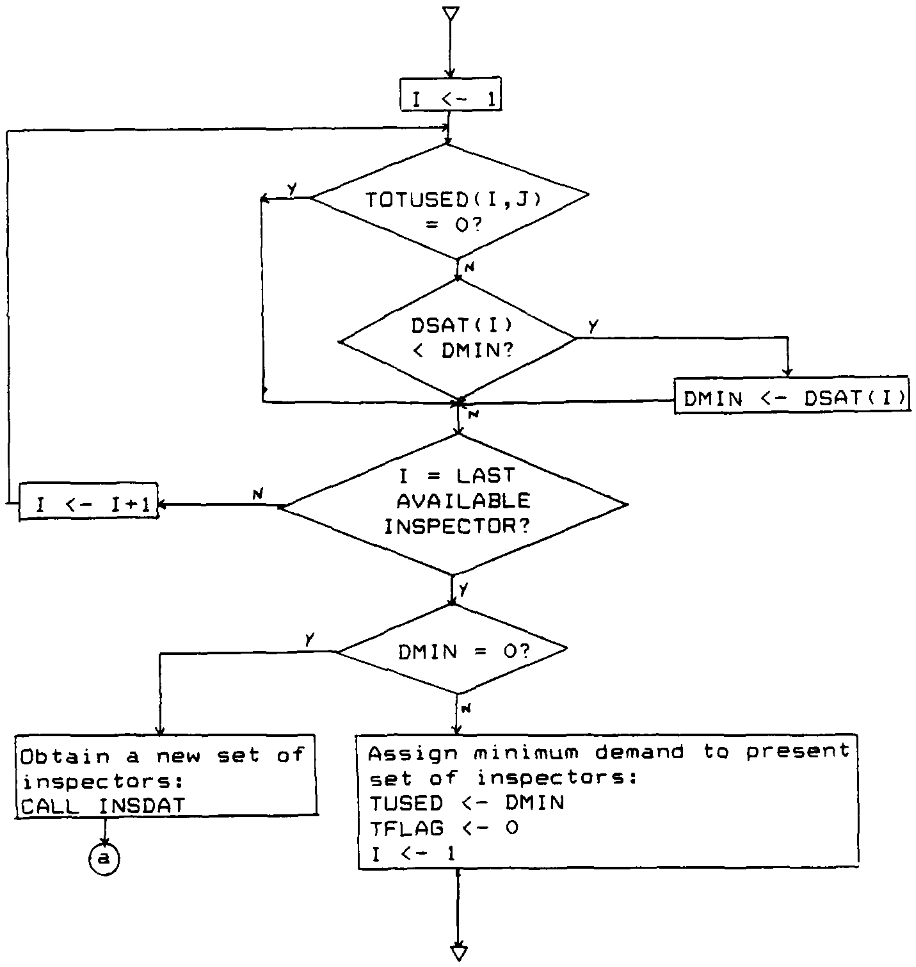
SUBROUTINE SLKDAT:

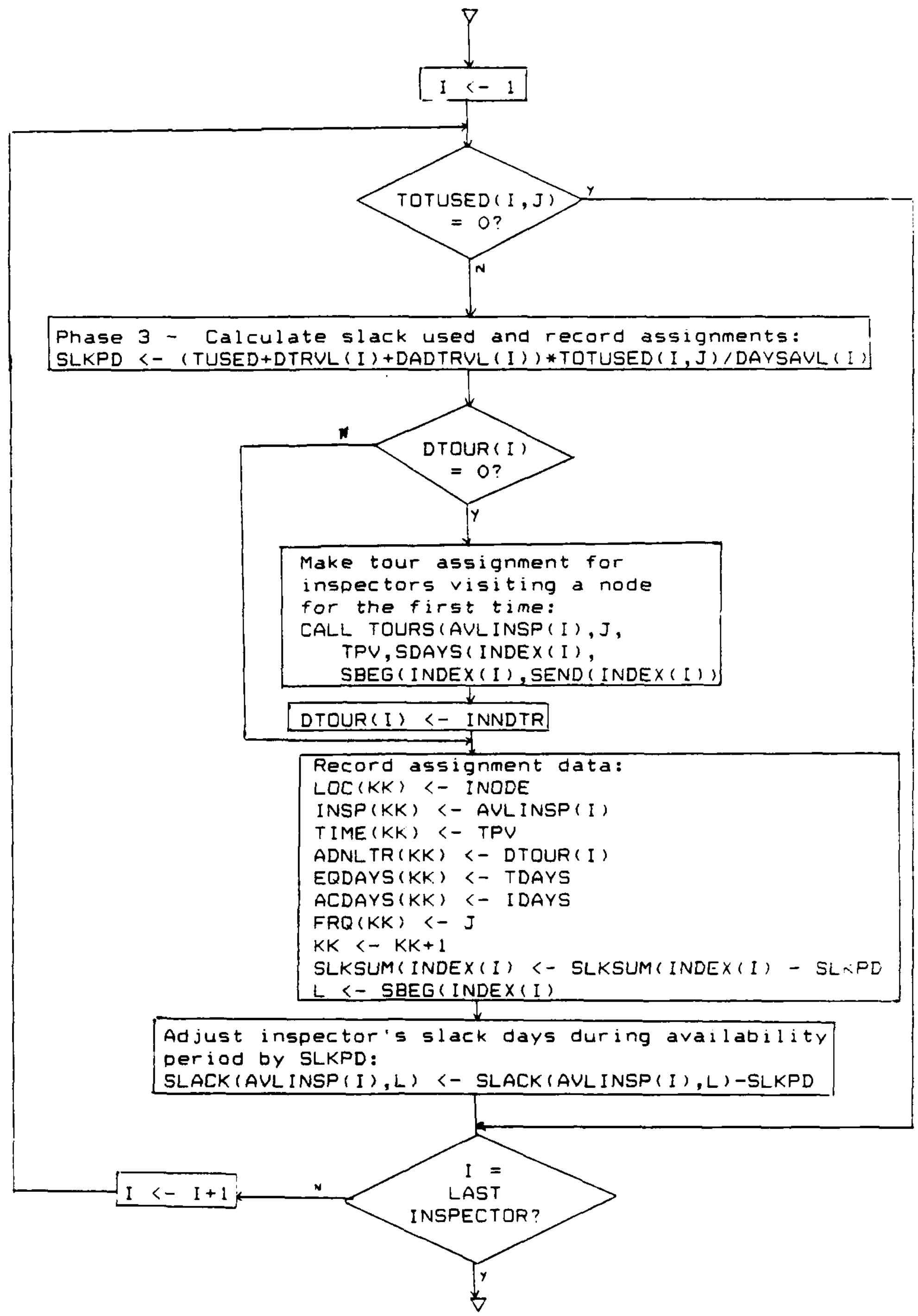


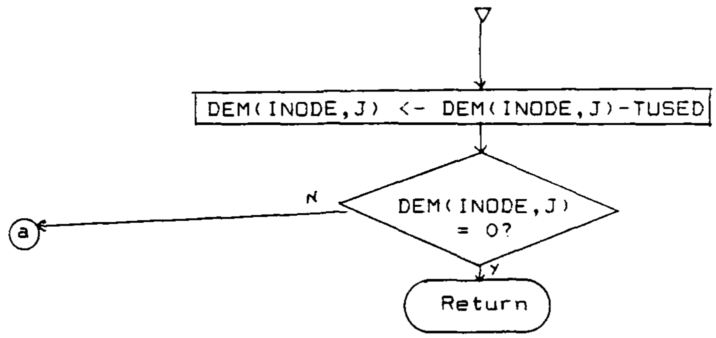
PROGRAM MODULE ASSIGN2(J):



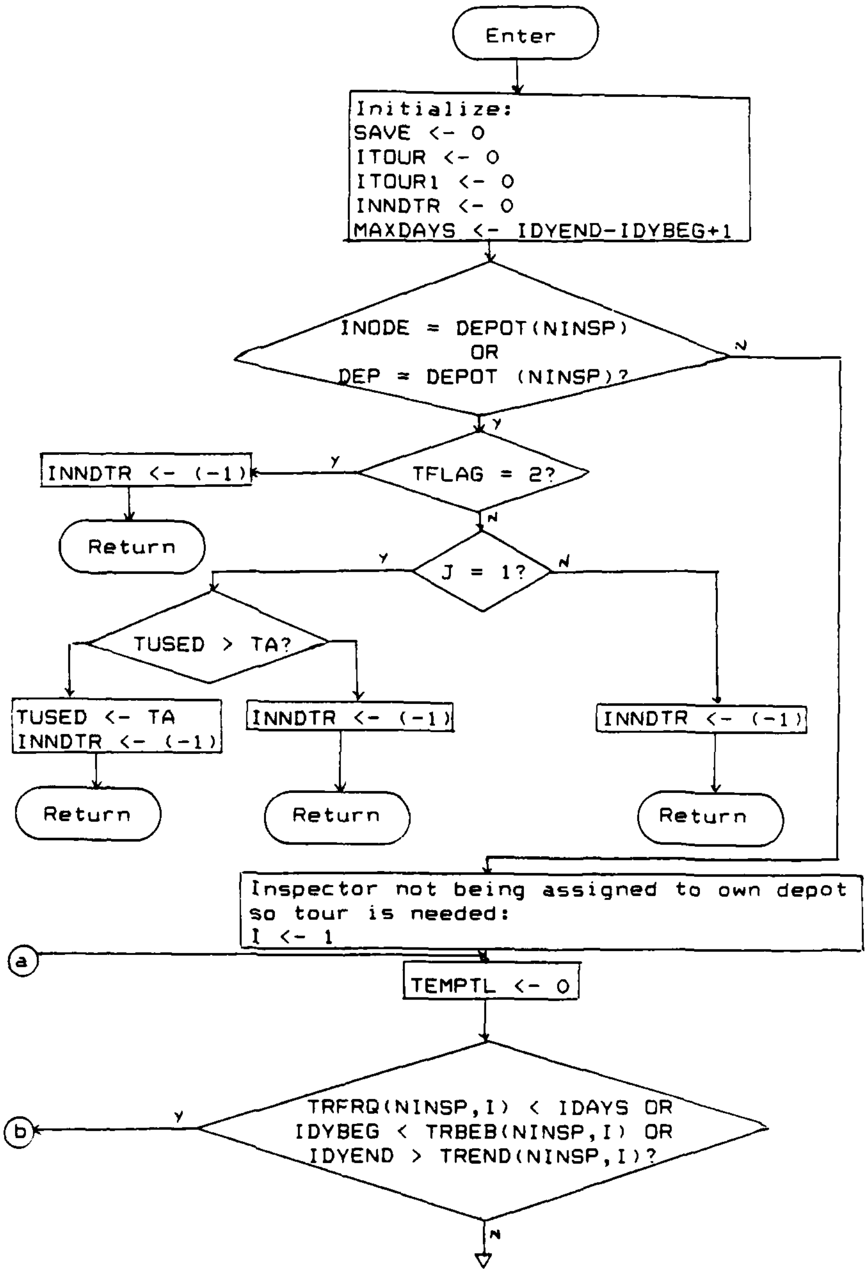

I <- 1
TOTUSED(I,J) = 0?
Y
N
Phase 2 - Determine amount of DMIN each inspector can satisfy:
TOTSLK <- (DMIN+DTRVL(I)+DADTRVL(I))*TOTUSED(I,J)
TRVSLK <- (DTRVL(I)+DADTRVL(I))*TOTUSED(I,J)
SLKAVL <- SLKSUM(INDEX(I))*DAYSAVL(I)
SLKAVL > TRVSLK?
Y
N
TOTSLK > SLKAVL?
N
Y
All demand satisfied:
DSAT(I) <- DMIN
All slack consumed:
DSAT(I) <- SLKSUM(INDEX(I)
*DAYSAVL(I)/T-TRVL
No demand satisfied:
DSAT(I) <- 0
SFLAG(INDEX(I) <- 1
I = LAST AVAILABLE INSPECTOR?
N
I <- I+1
Y

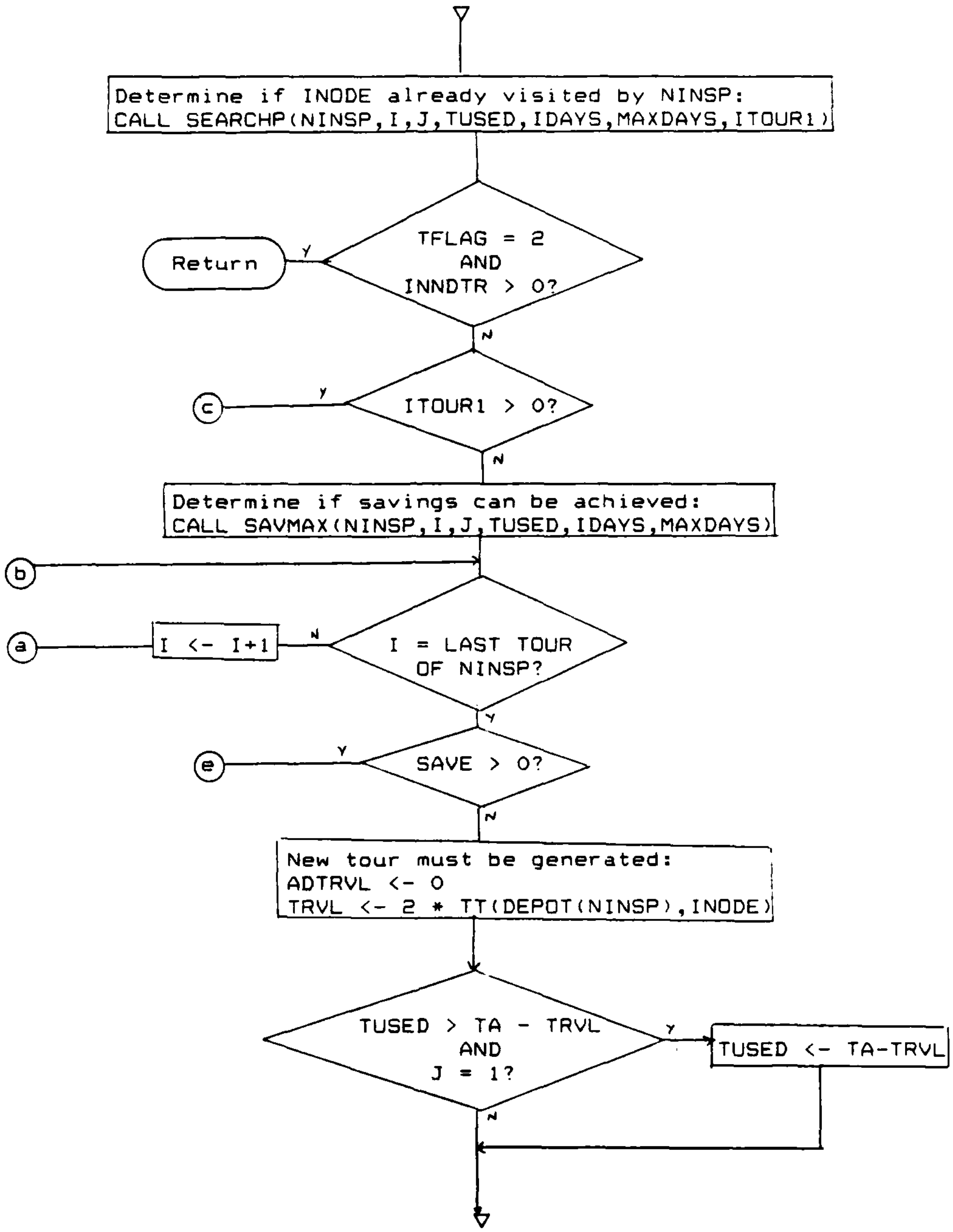

I <- 1
TOTUSED(I,J) = 0?
Y
N
DSAT(I) < DMIN?
Y
DMIN <- DSAT(I)
N
I = LAST AVAILABLE INSPECTOR?
N
I <- I+1
Y
DMIN = 0?
Y
N
Obtain a new set of inspectors: CALL INSDAT
a
Assign minimum demand to present set of inspectors:
TUSED <- DMIN
TFLAG <- 0
I <- 1

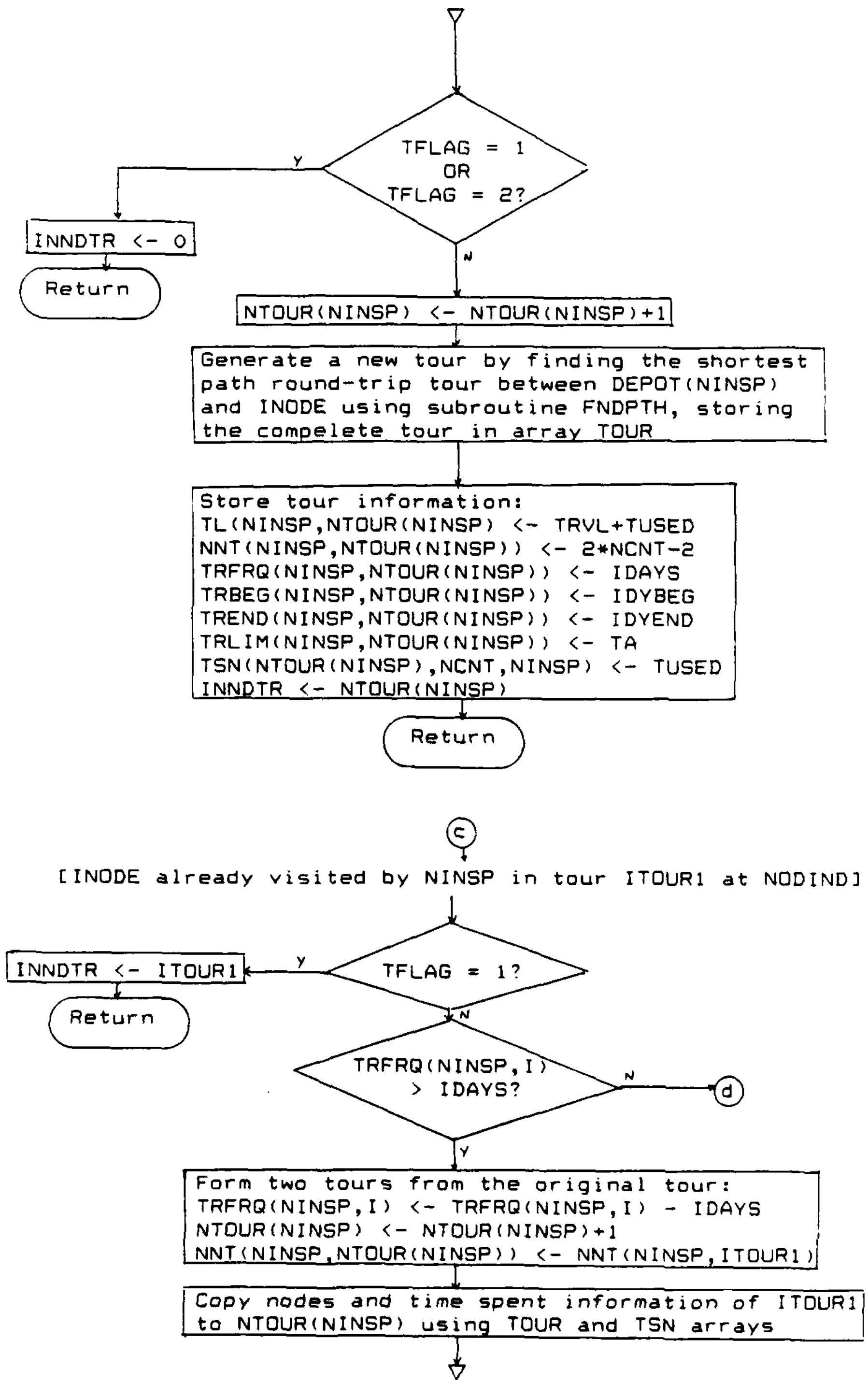

I <- 1
TOTUSED(I,J) = 0?
Y
N
Phase 3 - Calculate slack used and record assignments:
SLKPD <- (TUSED+DTRVL(I)+DADTRVL(I))*TOTUSED(I,J)/DAYSAVL(I)
DTOUR(I) = 0?
N
Y
Make tour assignment for inspectors visiting a node for the first time:
CALL TOURS(AVLINSP(I),J, TPV,SDAYS(INDEX(I), SBEG(INDEX(I),SEND(INDEX(I))
DTOUR(I) <- INNDTR
Record assignment data:
LOC(KK) <- INODE
INSP(KK) <- AVLINSP(I)
TIME(KK) <- TPV
ADNLTR(KK) <- DTOUR(I)
EQDAYS(KK) <- TDAYS
ACDAYS(KK) <- IDAYS
FRQ(KK) <- J
KK <- KK+1
SLKSUM(INDEX(I) <- SLKSUM(INDEX(I) - SLKPD
L <- SBEG(INDEX(I)
Adjust inspector's slack days during availability period by SLKPD:
SLACK(AVLINSP(I),L) <- SLACK(AVLINSP(I),L)-SLKPD
I = LAST INSPECTOR?
N
I <- I+1
Y

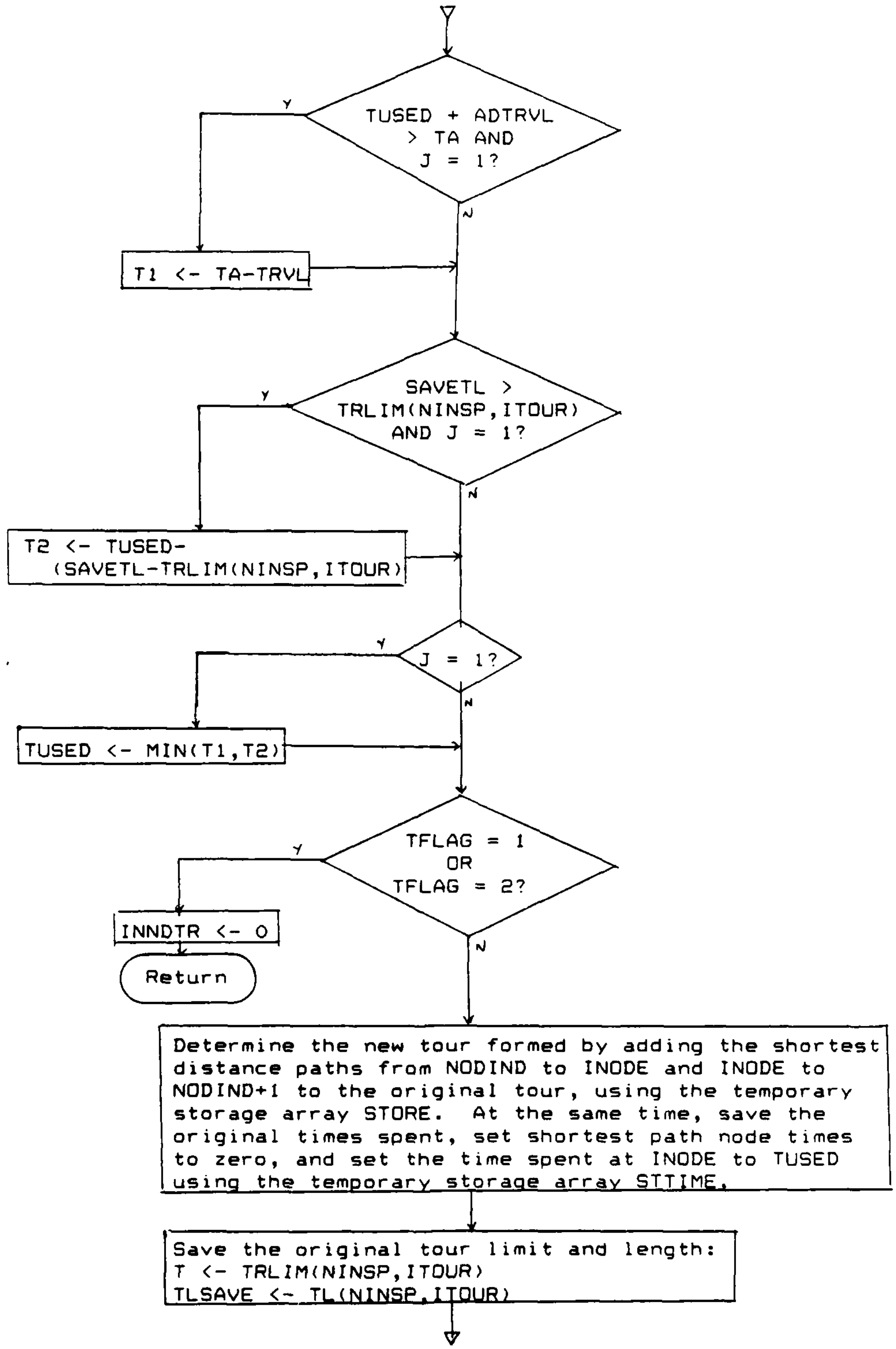

DEM(INODE,J) <- DEM(INODE,J)-TUSED
DEM(INODE,J) = 0?
N
Y
a
Return

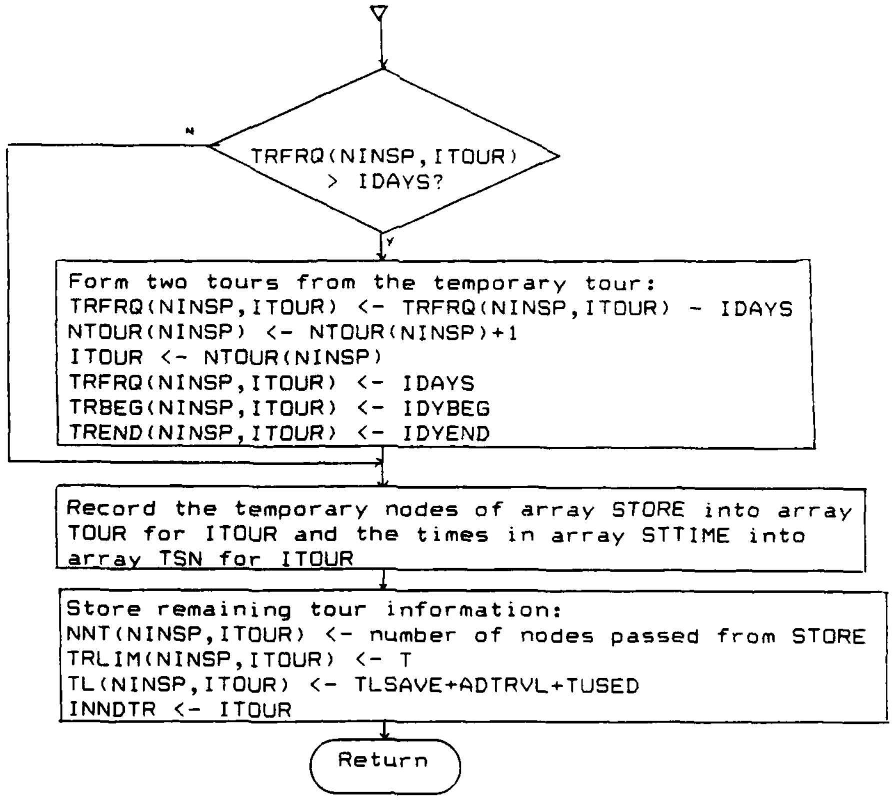
PROGRAM MODULE TOURS(NINSP,J,TUSED,IDAYS,IDYBEG,IDYEND):



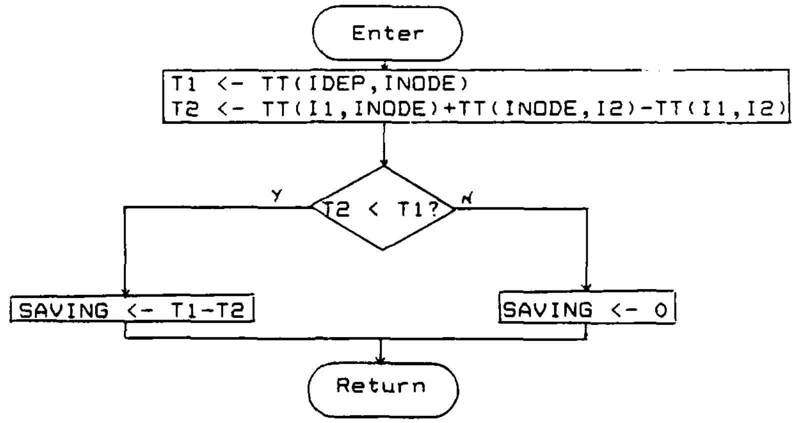

Determine if INODE already visited by NINSP:
CALL SEARCHP(NINSP,I,J,TUSED,IDAYS,MAXDAYS,ITOUR1)
TFLAG = 2
AND
INNDTR > 0?
Y
Return
N
ITOUR1 > 0?
Y
c
N
Determine if savings can be achieved:
CALL SAVMAX(NINSP,I,J,TUSED,IDAYS,MAXDAYS)
b
I = LAST TOUR
OF NINSP?
N
I <- I+1
a
Y
SAVE > 0?
Y
e
N
New tour must be generated:
ADTRVL <- 0
TRVL <- 2 * TT(DEPOT(NINSP),INODE)
TUSED > TA - TRVL
AND
J = 1?
Y
TUSED <- TA-TRVL
N


TFLAG = 1
OR
TFLAG = 2?
Y
INNDTR <- 0
Return
N
NTOUR(NINSP) <- NTOUR(NINSP)+1
Generate a new tour by finding the shortest path round-trip tour between DEPOT(NINSP) and INODE using subroutine FNDPTH, storing the compelete tour in array TOUR
Store tour information:
TL(NINSP,NTOUR(NINSP) <- TRVL+TUSED
NNT(NINSP,NTOUR(NINSP)) <- 2*NCNT-2
TRFRQ(NINSP,NTOUR(NINSP)) <- IDAYS
TRBEG(NINSP,NTOUR(NINSP)) <- IDYBEG
TREND(NINSP,NTOUR(NINSP)) <- IDYEND
TRLIM(NINSP,NTOUR(NINSP)) <- TA
TSN(NTOUR(NINSP),NCNT,NINSP) <- TUSED
INNDTR <- NTOUR(NINSP)
Return
c
[INODE already visited by NINSP in tour ITOUR1 at NODIND]
TFLAG = 1?
Y
INNDTR <- ITOUR1
Return
N
TRFRQ(NINSP,I)
> IDAYS?
N
d
Y
Form two tours from the original tour:
TRFRQ(NINSP,I) <- TRFRQ(NINSP,I) - IDAYS
NTOUR(NINSP) <- NTOUR(NINSP)+1
NNT(NINSP,NTOUR(NINSP)) <- NNT(NINSP,ITOUR1)
Copy nodes and time spent information of ITOUR1 to NTOUR(NINSP) using TOUR and TSN arrays

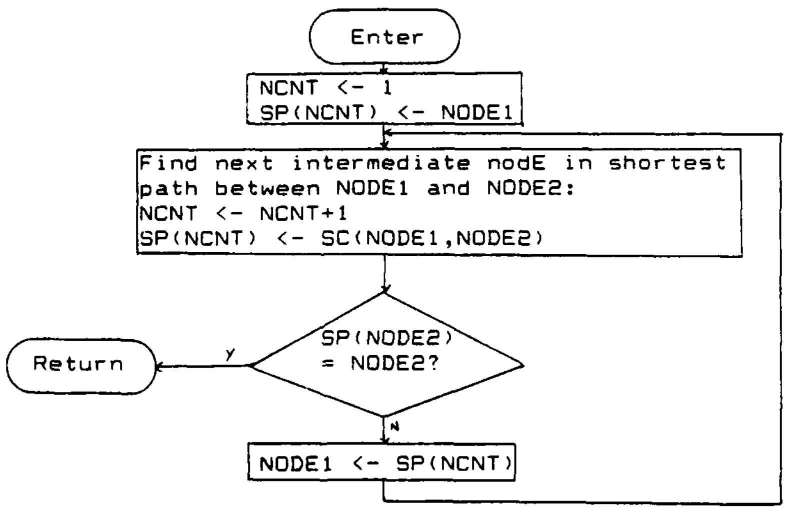

TUSED + ADTRVL > TA AND J = 1?
Y
N
T1 <- TA-TRVL
SAVETL > TRLIM(NINSP,ITOUR) AND J = 1?
Y
N
T2 <- TUSED-(SAVETL-TRLIM(NINSP,ITOUR)
J = 1?
Y
N
TUSED <- MIN(T1,T2)
TFLAG = 1 OR TFLAG = 2?
Y
N
INNDTR <- 0
Return
Determine the new tour formed by adding the shortest distance paths from NODIND to INODE and INODE to NODIND+1 to the original tour, using the temporary storage array STORE. At the same time, save the original times spent, set shortest path node times to zero, and set the time spent at INODE to TUSED using the temporary storage array STTIME.
Save the original tour limit and length:
T <- TRLIM(NINSP,ITOUR)
TLSAVE <- TL(NINSP,ITOUR)


N
TRFRQ(NINSP,ITOUR) > IDAYS?
Y
Form two tours from the temporary tour:
TRFRQ(NINSP,ITOUR) <- TRFRQ(NINSP,ITOUR) - IDAYS
NTOUR(NINSP) <- NTOUR(NINSP)+1
ITOUR <- NTOUR(NINSP)
TRFRQ(NINSP,ITOUR) <- IDAYS
TRBEG(NINSP,ITOUR) <- IDYBEG
TREND(NINSP,ITOUR) <- IDYEND
Record the temporary nodes of array STORE into array TOUR for ITOUR and the times in array STTIME into array TSN for ITOUR
Store remaining tour information:
NNT(NINSP,ITOUR) <- number of nodes passed from STORE
TRLIM(NINSP,ITOUR) <- T
TL(NINSP,ITOUR) <- TLSAVE+ADTRVL+TUSED
INNDTR <- ITOUR
Return

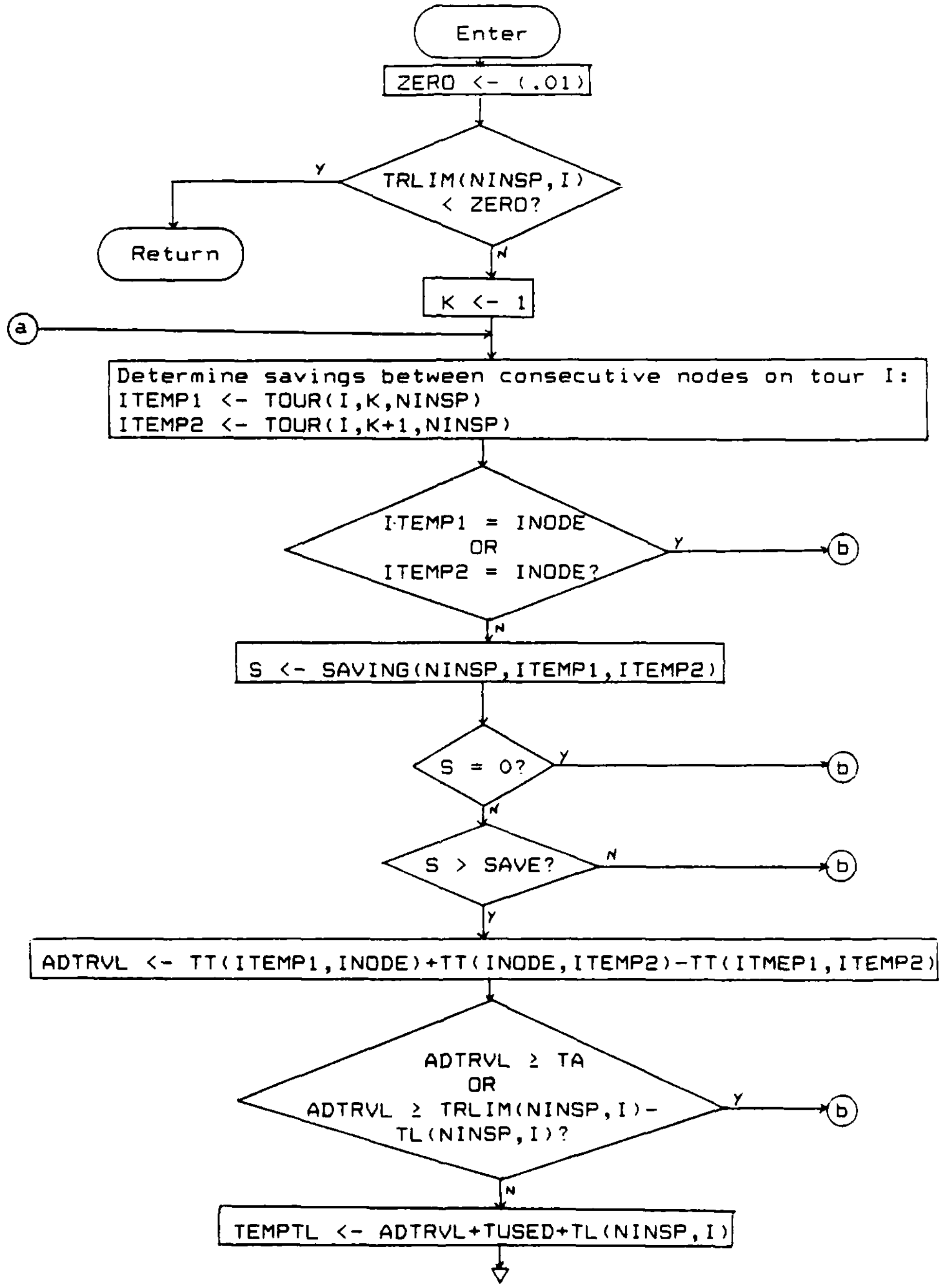
FUNCTION SAVING(NINSP,I1,I2):

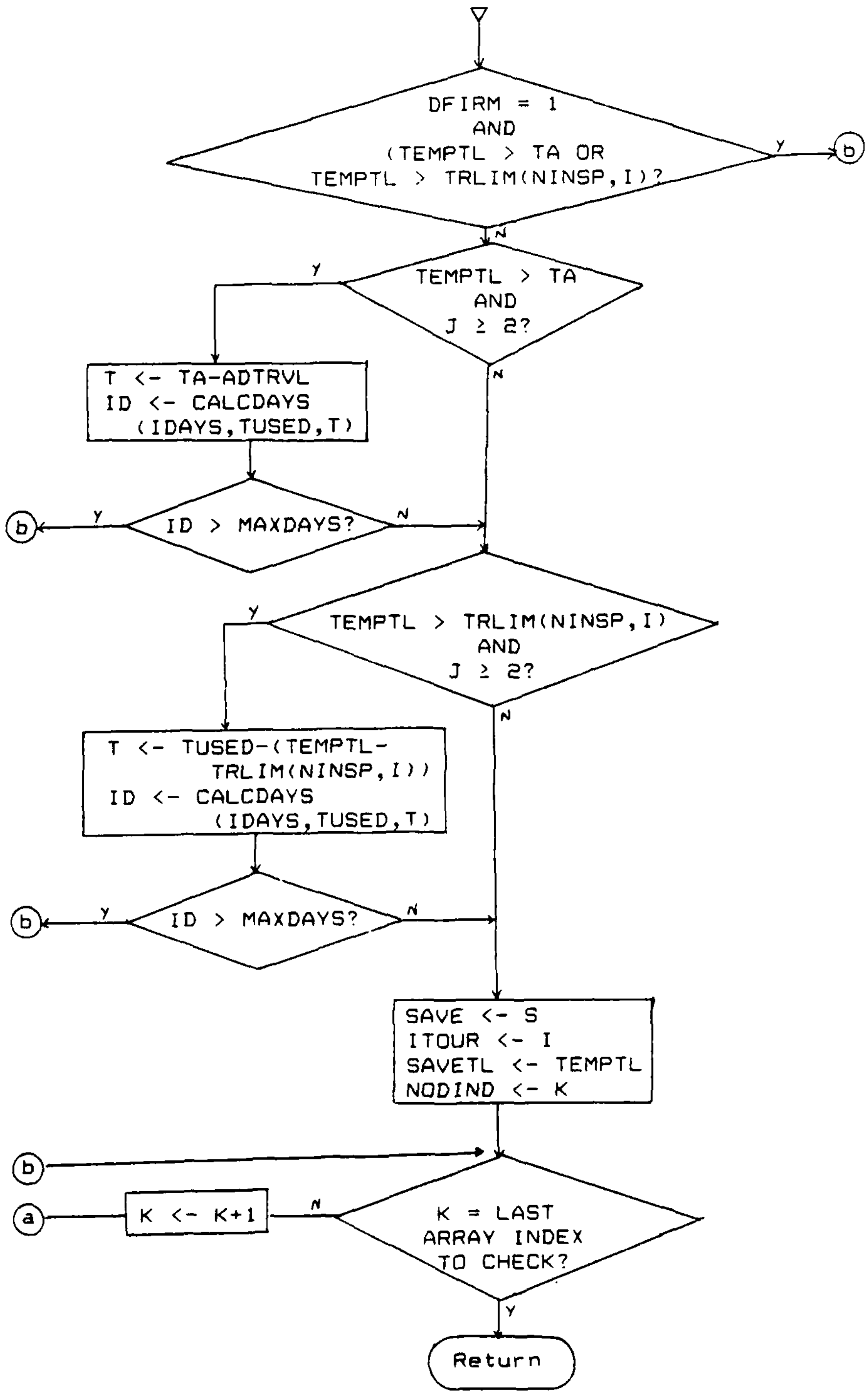


SUBROUTINE FNDPTH(NODE1,NODE2):

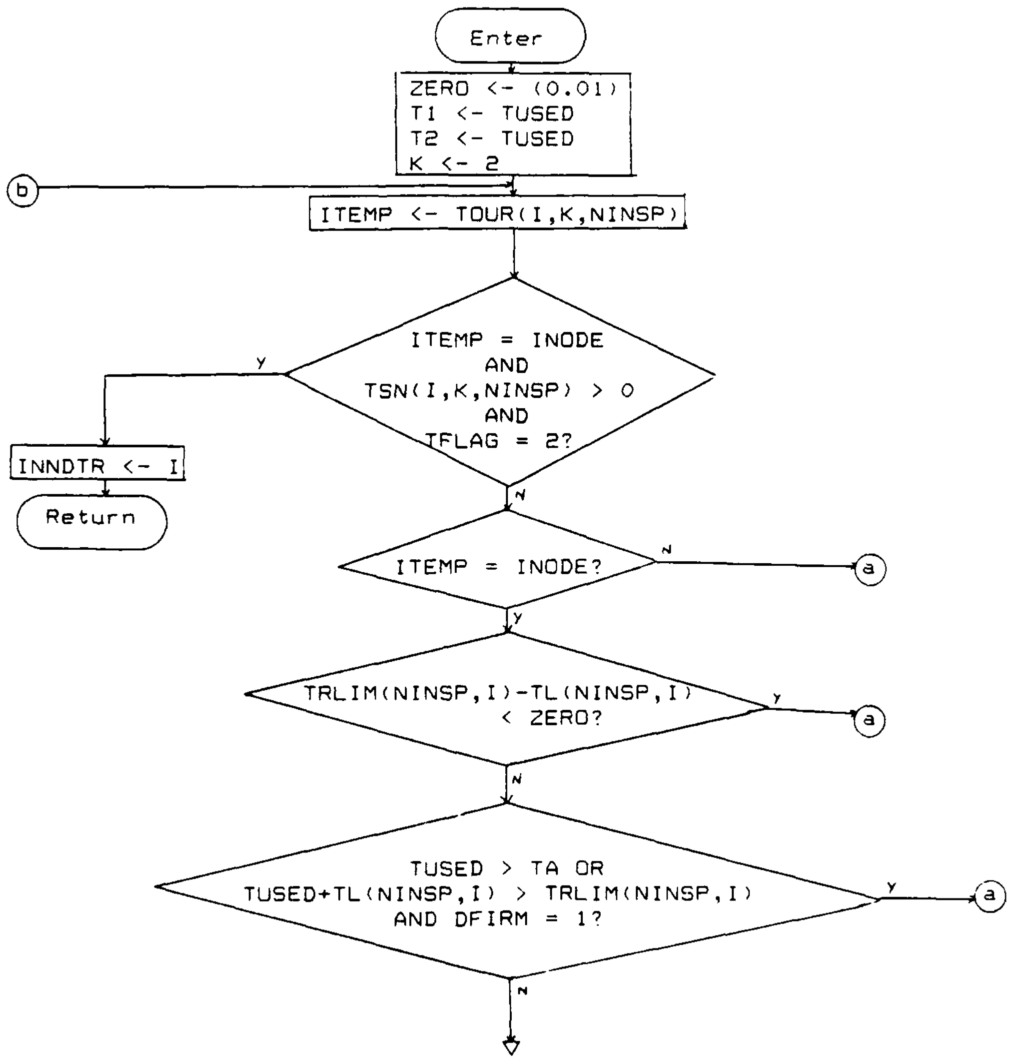


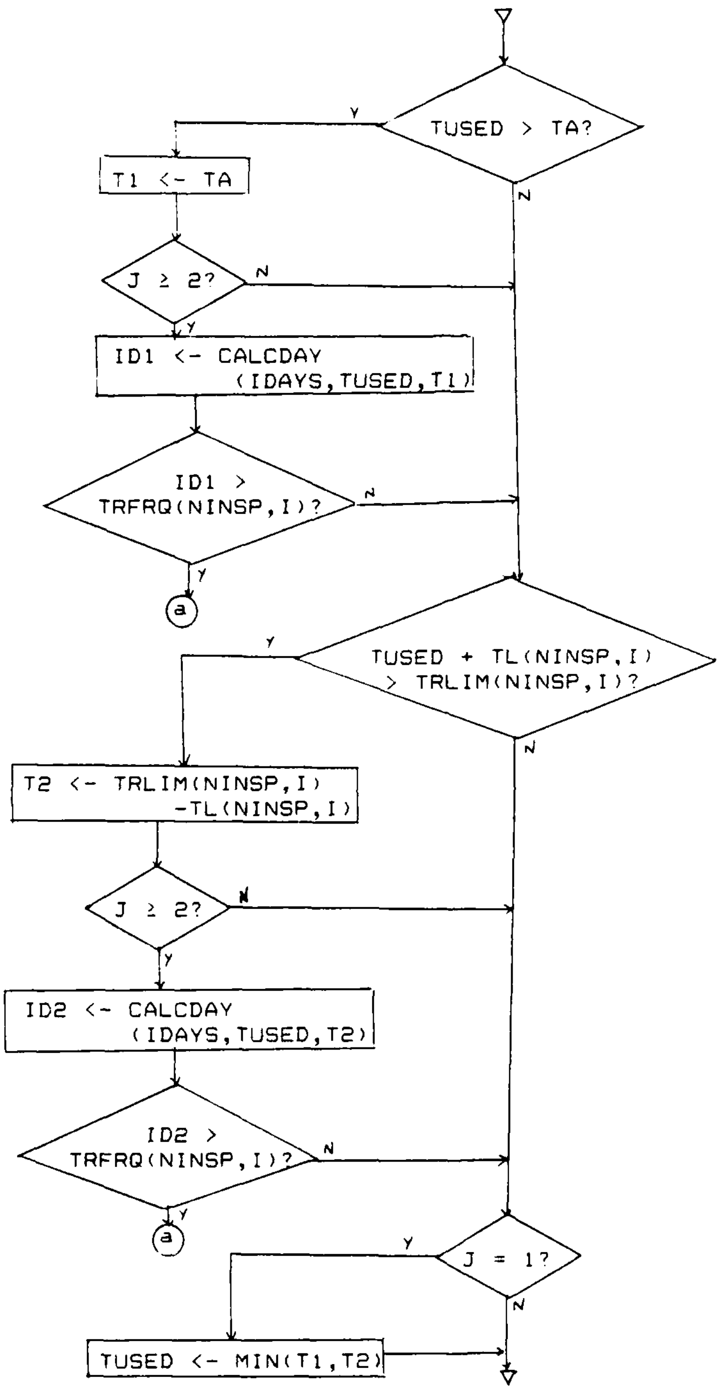
SUBROUTINE SAVMAX(NINSP,I,J,TUSED,IDAYS,MAXDAYS):

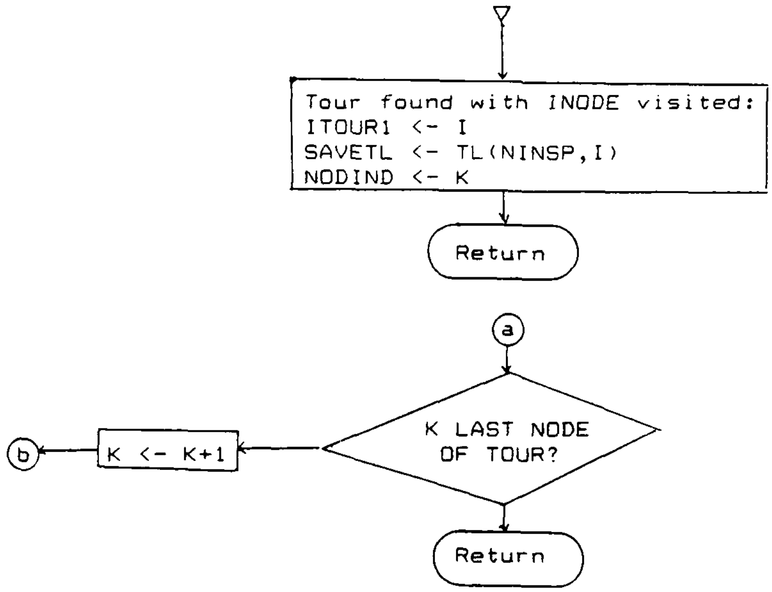



DFIRM = 1
AND
(TEMPTL > TA OR
TEMPTL > TRLIM(NINSP,I)?
Y
b
N
TEMPTL > TA
AND
J ≥ 2?
Y
N
T <- TA-ADTRVL
ID <- CALCDAYS
(IDAYS,TUSED,T)
ID > MAXDAYS?
Y
b
N
TEMPTL > TRLIM(NINSP,I)
AND
J ≥ 2?
Y
N
T <- TUSED-(TEMPTL-
TRLIM(NINSP,I))
ID <- CALCDAYS
(IDAYS,TUSED,T)
ID > MAXDAYS?
Y
b
N
SAVE <- S
ITOUR <- I
SAVETL <- TEMPTL
NODIND <- K
b
a
K <- K+1
N
K = LAST
ARRAY INDEX
TO CHECK?
Y
Return

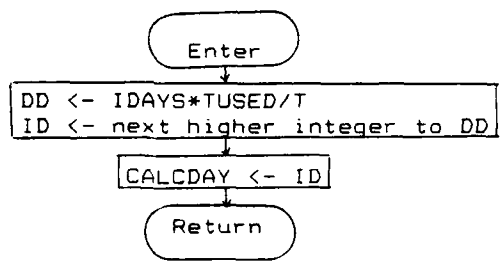
SUBROUTINE SEARCHP(NINSP,I,J,TUSED,IDAYS,MAXDAYS,ITOUR1):




TUSED > TA?
Y
N
T1 <- TA
J ≥ 2?
N
Y
ID1 <- CALCDAY
(IDAYS,TUSED,T1)
ID1 >
TRFRQ(NINSP,I)?
N
Y
a
TUSED + TL(NINSP,I)
> TRLIM(NINSP,I)?
Y
N
T2 <- TRLIM(NINSP,I)
-TL(NINSP,I)
J ≥ 2?
N
Y
ID2 <- CALCDAY
(IDAYS,TUSED,T2)
ID2 >
TRFRQ(NINSP,I)?
N
Y
a
J = 1?
Y
N
TUSED <- MIN(T1,T2)



FUNCTION CALCDAY(IDAYS,TUSED,T):



APPENDIX B: COMPUTER CODE

```
****  ISTAR PROGRAM MODULE - MAIN PROGRAM  ****
*
      PROGRAM ISTAR
$INCLUDE: 'COMMONS.INS'
C
C   INPUT DATA FOR 19 NODE, 2-DEPOT NETWORK OF FIGURE 4.1
C
C   PARAMETERS SET FOR 252 TOTAL WORKING DAYS, 218 DAYS OF INSPECTOR
C   AVAILABILITY, WITH MAXIMUM 6.5 HOURS PER DAY

C
C   INITIALIZE FREQUENCY PERIODS
C
      FP(1)=252
      FP(2)=52
      FP(3)=26
      FP(4)=12
      FP(5)=4
      FP(6)=2
      FP(7)=1
C
C   INITIALIZE GIVEN PARAMETERS
C
      NN=19
      NUMDEP=2
      TDA=218
      DR=252
      TAMAX=6.5
      P=7
C
C   INITIALIZE DEPOTS
C
      DEPLIST(1)=1
      DEPLIST(2)=14
      DEPLIST(NUMDEP+1)=NN+1
C
C   INITIALIZE DAILY TIME AVAILABLE FOR EVERY POSSIBLE INSPECTOR
C
      DO 7 I=1,MAXINSP
         DO 6 J=1,MAXTIME
            SLACK(I,J)=TAMAX
6        CONTINUE
7     CONTINUE
C
C   INITIALIZE ARRAY COUNTERS
C
      KK=1
      KKCNT=1
      II=0
```

```
C
C   OPEN DATA AND REPORT FILES
C
        OPEN (1,FILE=' ',STATUS='OLD')
        OPEN (2,FILE=' ',STATUS='OLD')
        OPEN (4,FILE=' ',STATUS='NEW')
        OPEN (5,FILE=' ',STATUS='NEW')
C
C   READ INPUT DATA:  SHORTEST DISTANCE TIMES, TT; DEMAND, DEM; AND
C   SHORTEST CHAIN MATRIX, SC
C
        READ (1,*) ((TT(I,J), J=1,NN), I=1,NN)
        READ (1,*) ((SC(I,J), J=1,NN), I=1,NN)
        READ (2,*) ((DEM(I,J), J=1,P), I=1,NN)
C
C   PERFORM DAILY AND NON-DAILY DEMAND ASSIGNMENTS FOR EVERY DEPOT BLOCK
C   DEFINED BETWEEN NNBEG AND NNEND
C
        DO 50 I=1,NUMDEP
           DEP=DEPLIST(I)
           NNBEG=DEPLIST(I)
           NNEND=DEPLIST(I+1)-1
           CALL DAILY(NNBEG,NNEND)
           CALL CYCLIC(NNBEG,NNEND)
50      CONTINUE
C
C   OUTPUT FINAL SLACK SUMMARY, TOURS, AND ASSIGNMENTS TO FILE 4
C
        WRITE (4,90)
90       FORMAT (///,1X,'************ FINAL SLACK SUMMARY ************',/)
        CALL SLKOUT
        CALL TROUT
        CALL ADOUT
C
C   OUTPUT DATA AND ASSIGNMENT STATISTICS TO FILE 5
C
        CALL STATS
C
        STOP
        END
```

```
****  DAILY PROGRAM MODULE ****
*
      SUBROUTINE DAILY(NNBEG,NNEND)
$INCLUDE: 'COMMONS.INS'
      WRITE (*,5) DEP
5     FORMAT (///,1X,'************** DAILY DEMAND ITERATIONS FOR ',
     +         'DEPOT ',I2,' *************',//)
C
C  MAKE ASSIGNMENTS TO ALL NODES IN BLOCK WITH POSITIVE DAILY DEMAND
C
10    DO 20 I=NNBEG,NNEND
         DAY=1
         TDD=DEM(I,1)
         IF (TDD.EQ.0) GOTO 20
         NNV=NNV+1
         NV(NNV)=I
         INODE=I
         CALL ASSIGN
         DEM(I,1)=0.
20    CONTINUE
C
C  OUTPUT LISTING OF DAILY ASSIGMENTS MADE AND UPDATED SLACK SUMMARY
C
      IF (KKCNT.EQ.KK) GOTO 29
      WRITE (4,25) DEP
25    FORMAT (//,5X,'SUMMARY OF DAILY ASSIGNMENTS FOR DEPOT ',I2)
      CALL LOCOUT
      KKCNT=KK
29    WRITE (4,30) DEP
30    FORMAT (//,5X,'SLACK SUMMARY AFTER DAILY ASSIGNMENTS FOR DEPOT ',
     +         I2)
      CALL SLKOUT
C
C  FIND INSPECTORS WITH SLACK TIME COVERAGE OVER ALL WORKING DAYS
C
      CALL INSDAT
C
C  MAKE ASSIGNMENTS FOR ALL NODES VISITED DAILY WITH NON-DAILY DEMAND
C
      DO 150 J=2,P
         DO 100 I=1,NNV
            INODE=NV(I)
            IF (DEM(INODE,J).EQ.0) GOTO 100
            CALL USESLK(J)
100      CONTINUE
150   CONTINUE
C
C  OUTPUT LISTING OF ASSIGNMENTS MADE AND UPDATED SLACK SUMMARY
C
      IF (KKCNT.EQ.KK) GOTO 155
      WRITE (4,165) DEP
      CALL LOCOUT
      KKCNT=KK
155   WRITE (4,160) DEP
```

```
160   FORMAT (//,5X,'SLACK AFTER SATISFYING CYCLIC DEMANDS OF NODES ',
     +         'VISITED DAILY FOR DEPOT ',I2)
      CALL SLKOUT
165   FORMAT (//,5X,'NON-DAILY ASSIGNMENTS FOR NODES VISITED ',
     +         'DAILY FOR DEPOT ',I2)
      RETURN
      END
```

```
****  CYCLIC PROGRAM MODULE ****
*
      SUBROUTINE CYCLIC(NNBEG,NNEND)
$INCLUDE: 'COMMONS.INS'
      WRITE (*,10) DEP
10    FORMAT (///,1X,'************ CYCLIC DEMAND ITERATIONS FOR NODES
     +        'OF DEPOT ',I2,'************',//)
C
C  MAKE NON-DAILY INSPECTOR ASSIGNMENTS TO ALL NODES OF DEPOT BLOCK
C  FOR EVERY FREQUENCY PERIOD JJ
C
      DO 100 JJ=2,P
         DO 50 I=NNBEG,NNEND
            IF (DEM(I,JJ).EQ.0.) GOTO 50
            NNV=NNV+1
            NV(NNV)=I
            INODE=I
            CALL USESLK(JJ)
50       CONTINUE
C
C  FOR EVERY NODE VISITED IN PERIOD JJ WITH LOWER FREQUENCY DEMAND
C  REQUIREMENTS, MAKE REMAINING NON-DAILY INSPECTOR ASSIGNMENTS
C
         DO 80 J=JJ+1,P
            DO 70 I=1,NNV
               INODE=NV(I)
               IF (DEM(INODE,J).EQ.0) GOTO 70
               CALL USESLK(J)
70          CONTINUE
80       CONTINUE
100   CONTINUE
C
C  OUTPUT LISTING OF ASSIGNMENTS MADE AND UPDATED SLACK SUMMARY
C
      IF (KKCNT.EQ.KK) GOTO 105
      WRITE (4,115)
      CALL LOCOUT
      KKCNT=KK
105   WRITE (4,110) DEP
      CALL SLKOUT
      RETURN
110   FORMAT (//,5X,'SLACK AFTER SATISFYING ALL CYCLIC DEMANDS FOR ',
     +        'NODES OF DEPOT ',I2)
115   FORMAT (//,5X,'ASSIGNMENTS MADE TO INSPECTORS WITH ',
     +        'SUFFICIENT SLACK TIME',/)
      END
```

```
****   PROGRAM MODULE INSDAT   ****
*
      SUBROUTINE INSDAT
$INCLUDE: 'COMMONS.INS'
      ZERO=.01
      NEXT=0
C
C  INITIALIZE ARRAY REMAIN TO THE NUMBER OF VISITS REQUIRED IN
C  EACH FREQUENCY PERIOD
C
      DO 10 L=1,P
         REMAIN(L)=FP(L)
10    CONTINUE
C
C  IF TOTAL INSPECTORS ASSIGNED IS 0, ADD FIRST INSPECTOR WITH
C  AVAILABILITY BEGINNING ON DAY 1
C
      IF (TI.EQ.0) THEN
         NEXTSV=0
         II=1
         CALL ADDINS
      ENDIF
C
C  FIND THE MOST RECENTLY ADDED INSPECTOR WITH POSITIVE SLACK TIME
C
20    DO 100 K=1,II-1
      I=II-K
      INSPCNT=0
      IDAYS=SEND(I)-SBEG(I)+1
      IF (SLKSUM(I).LT.ZERO.OR.SFLAG(I).GT.0) GOTO 100
C
C  INSPECTOR FOUND
C
      INSPCNT=INSPCNT+1
      AVLINSP(INSPCNT)=SINSP(I)
      DAYSAVL(INSPCNT)=SDAYS(I)
      HRSAVL(INSPCNT)=SLKSUM(I)
      INDEX(INSPCNT)=I
C
C  IF INSPECTOR'S SLACK COVERS TOTAL WORKING DAYS, THEN GO TO CALCULATE
C  EQUIVALENT NUMBER OF DAYS AVAILABLE IN EACH FREQUENCY PERIOD
C
      IF (IDAYS.GE.DR) GOTO 200
C
C  OTHERWISE, DETERMINE FIRST DAY OF AVAILABILITY FOR NEXT INSPECTOR
C
      IF (SBEG(I).EQ.1) THEN
         NEXT=DR
      ELSE
         NEXT=SBEG(I)-1
      ENDIF
```

```
C
C  FIND ANOTHER INSPECTOR PROVIDING CONTINOUS COVERAGE WITH THE LAST
C  INSPECTOR FOUND
C
      DO 90 L=1,II-1
         IF (L.EQ.K) GOTO 90
         J=II-L
         JDAYS=SEND(J)-SBEG(J)+1
         IF (SLKSUM(J).LT.ZERO.OR.SEND(J).NE.NEXT.OR.SFLAG(J).GT.0)
     +       GOTO 90
C
C  INSPECTOR FOUND
C
         INSPCNT=INSPCNT+1
         AVLINSP(INSPCNT)=SINSP(J)
         DAYSAVL(INSPCNT)=SDAYS(J)
         HRSAVL(INSPCNT)=SLKSUM(J)
         INDEX(INSPCNT)=J
         IDAYS=IDAYS+JDAYS
C
C  IF ALL INSPECTORS' SLACK COVERS TOTAL WORKING DAYS, THEN GO TO
C  CALCULATE EQUIVALENT NUMBER OF DAYS AVAILABLE IN EACH PERIOD
C
         IF (IDAYS.GE.DR) GOTO 200
C
C  OTHERWISE, CALCULATE TOTAL SLACK HOURS PROVIDED BY PRESENT SET OF
C  INSPECTORS AND DETERMINE FIRST DAY OF AVAILABILITY FOR ANOTHER
C  INSPECTOR
C
         IF (SBEG(J).EQ.1) THEN
            NEXT=DR
         ELSE
            NEXT=SBEG(J)-1
         ENDIF
90    CONTINUE
100   CONTINUE
C
C  IF NO SET OF INSPECTORS FOUND WITH COMPLETE COVERAGE, ADD ANOTHER
C  INSPECTOR WITH AVAILABILITY BEGINNING AT NEXTSV AND RETRY PROCESS
C
      CALL ADDINS
      GOTO 20
```

```
C
C  CALCULATE EQUIVALENT NUMBER OF DAYS AVAILABLE FOR EACH INSPECTOR
C  DURING EVERY FREQUENCY PERIOD WITH INSPECTOR WITH MOST TOTAL
C  AVAILABILITY RECEIVING PRIORITY:
C
C       EQUIV DAYS = DAYS AVAILABLE*DAYS REQUIRED/TOTAL WORKING DAYS
C
200    IF (INSPCNT.GT.1) CALL SORTIN
          DO 250  I=1,INSPCNT
          DO 240 J=1,P
          D=DAYSAVL(I)
          F=FP(J)
          T=DR
          X=D*F/T
C
C  ROUND EQUIVALENT NUMBER OF DAYS UP TO NEXT INTEGER, IF NECESSARY
C
          IX=NINT(X)
          IF (IX.GE.X) GOTO 220
          IX=IX+1
C
C  ASSIGN TOTUSED(I,J) WITH NUMBER OF DAYS OF INSPECTOR I IN FREQUENCY
C  PERIOD J:
C
C       TOTUSED(I,J) = MIN(EQUIV DAYS,REMAIN(J))
C
C  WHERE REMAIN(J) IS THE NUMBER OF DAYS REMAINING TO BE ASSIGNED IN
C  FREQUENCY PERIOD J
C
220       IF (I.EQ.1) THEN
             TOTUSED(I,J)=IX
             REMAIN(J)=REMAIN(J)-TOTUSED(I,J)
          ELSE
             TOTUSED(I,J)=MIN(IX,REMAIN(J))
             REMAIN(J)=REMAIN(J)-TOTUSED(I,J)
          ENDIF
240       CONTINUE
250    CONTINUE
C
C  OUTPUT INSPECTOR DATA
C
       CALL IDATOUT
       RETURN
       END
C
```

```
C
      SUBROUTINE ADDINS
C
C  ADD A NEW INSPECTOR WITH AVAILABILITY BEGINNING ON DAY NEXTSV
C
$INCLUDE: 'COMMONS.INS'
C
C  INCREASE INSPECTOR COUNT AND ASSIGN INSPECTOR TO CURRENT DEPOT
C
      TI=TI+1
      DEPOT(TI)=DEP
      WRITE (*,'(1X,A,I2)') ' ADDING INSPECTOR NO ', TI
C
C  IF NUMBER OF INSPECTOR AVAILABLE DAYS LESS THAN TOTAL WORKING DAYS,
C  ASSIGN IDLE TIME AT THE END OF AVAILABLITY PERIOD
C
      MAXIDLE=DR-TDA
      IF (NEXTSV.EQ.0) NEXTSV=NEXTSV+1
      IF (MAXIDLE.EQ.0) RETURN
      NEXTSV=NEXTSV+TDA-MAXIDLE
      IF (NEXTSV.GT.DR) NEXTSV=NEXTSV-DR
      IDY=NEXTSV
      DO 100 I=1,MAXIDLE
         IF (IDY.GT.DR) IDY=1
         SLACK(TI,IDY)=0.
         IDY=IDY+1
100   CONTINUE
      CALL SLKDAT
      RETURN
      END
```

```
      SUBROUTINE SORTIN
C
C  SORT INSPECTOR SET IN INCREASING ORDER OF THEIR TOTAL AVAILABLE HOURS
C
$INCLUDE: 'COMMONS.INS'
      DO 100 I=1,INSPCNT
         DO 50 K=I+1,INSPCNT
            IF (HRSAVL(I)*DAYSAVL(I).LT.HRSAVL(K)*DAYSAVL(K)) THEN
               H=HRSAVL(I)
               ID=DAYSAVL(I)
               IND=INDEX(I)
               IINS=AVLINSP(I)
               IF=FDAY(I)
               IL=LDAY(I)
               HRSAVL(I)=HRSAVL(K)
               DAYSAVL(I)=DAYSAVL(K)
               INDEX(I)=INDEX(K)
               AVLINSP(I)=AVLINSP(K)
               FDAY(I)=FDAY(K)
               LDAY(I)=LDAY(K)
               HRSAVL(K)=H
               DAYSAVL(K)=ID
               INDEX(K)=IND
               AVLINSP(K)=IINS
               FDAY(K)=IF
               LDAY(K)=IL
            ENDIF
50       CONTINUE
100   CONTINUE
      RETURN
      END
```

```
****  PROGRAM MODULE ASSIGN1   ****
*
      SUBROUTINE ASSIGN1
$INCLUDE: 'COMMONS.INS'
C
C  INITIALIZE ASSIGNMENT VARIABLES
C
      ICNT=1
      DAY=1
      LAST=0
      DFIRM=0
      DEMAND=TDD
C
C  OBTAIN FIRST INSPECTOR TO RECEIVE ASSIGNMENT
C  DEMAND VALUE RETURNED IS AMOUNT OF TDD INSPECTOR CAN SATISFY
C
      CALL SLKINSP(DEMAND)
C
C  UPDATE TDD TO AMOUNT OF DEMAND REMAINING TO BE ASSIGNED
C
      TDD=TDD-DEMAND
C
C  ASSIGNMENT PROCESS FOR AN INSPECTOR UNTIL ICNT > DAYS AVAILABLE OR
C  DAY > LAST WORKING DAY
C
20    FIRST=DAY
25    IF (ICNT.LE.DA) THEN
         IF (DAY.GT.DR.AND.TDD.GT.0) THEN
            LAST=DR
            GOTO 50
         ELSEIF (DAY.GT.DR.AND.TDD.EQ.0) THEN
            LAST=DR
            GOTO 100
         ELSE
            SLACK(NI,DAY)=SLACK(NI,DAY)-(DEMAND+TRVL+ADTRVL)
            DAY=DAY+1
            ICNT=ICNT+1
         ENDIF
      ELSE
         IF (DAY.GT.DR.AND.TDD.GT.0) THEN
            LAST=DR
            GOTO 200
         ELSEIF (DAY.GT.DR.AND.TDD.EQ.0) THEN
            LAST=DR
            GOTO 100
         ELSE
            LAST=DAY-1
            GOTO 150
         ENDIF
      ENDIF
      GOTO 25
```

```
C  LAST DAY REACHED BEFORE ALL INSPECTOR AVAILABILITY USED AND MORE
C  DEMAND MUST BE SATISFIED SO RECORD ASSIGNMENT TO LAST WORKING DAY
C
50     CALL LOCDAT(DEMAND)
C
C  CONTINUE ASSIGNING INSPECTOR FROM DAY 1 WITH NEW AMOUNT OF DEMAND
C  RETURNED FROM CALL TO TOURS AND ADJUST DEMAND REMAINING
C
       DFIRM=0
       DEMAND=TDD
       TFLAG=0
       CALL TOURS(NI,1,DEMAND,DA-ICNT,1,DA-ICNT)
       TDD=TDD-DEMAND
       DAY=1
       GOTO 20
C
C  LAST DAY REACHED AND NO MORE DEMAND REMAINING SO RECORD ASSIGNMENT,
C  UPDATE SLACK, AND RETURN
C
100    CALL LOCDAT(DEMAND)
       CALL SLKDAT
       RETURN
C
C  INSPECTOR AVAILABILITY USED UP BEFORE REACHING LAST DAY SO RECORD
C  ASSIGNMENT AND UPDATE SLACK
C
150    CALL LOCDAT(DEMAND)
       CALL SLKDAT
C
C  FIND ANOTHER INSPECTOR WITH AVAILABILITY BEGINNING WHERE LAST LEFT
C  OFF WHO CAN SATISFY EXACT SAME AMOUNT OF DEMAND AS PREVIOUS INSPECTOR
C
       DFIRM=1
       CALL SLKINSP(DEMAND)
       ICNT=1
       DAY=LAST+1
       GOTO 20
C
C  LAST DAY OF INSPECTOR AVAILABILITY REACHED ON LAST DAY SO RECORD
C  INSPECTOR ASSIGNMENT AND UPDATE SLACK
C
200    CALL LOCDAT(DEMAND)
       CALL SLKDAT
C
C  FIND ANOTHER INSPECTOR WITH AVAILABILITY BEGINNING ON DAY 1 TO
C  SATISFY AMOUNT OF DEMAND RETURNED FROM CALL TO TOURS AND ADJUST
C  DEMAND REMAINING
C
       DFIRM=0
       DEMAND=TDD
       CALL SLKINSP(DEMAND)
       TDD=TDD-DEMAND
       ICNT=1
       DAY=1
```

```
      GOTO20
      END
C
C
C
      SUBROUTINE SLKINSP(DEMAND)
C
C  FIND AN INSPECTOR WITH SUFFICIENT SLACK TIME FOR TRAVEL TO INODE
C  WITH AVAILABILITY BEGINNING ON DAY SPECIFIED BY LAST
C
$INCLUDE: 'COMMONS.INS'
      IF (IDEPOT.EQ.0) IDEPOT=DEP
C
C  IF TOTAL INSPECTORS IS 0, ASSIGN FIRST INSPECTOR WITH FIRST DAY
C  OF AVAILABILITY ON DAY 1
C
      IF (TI.EQ.0) THEN
         NEXTSV=0
         IBEG=1
         GOTO20
      ENDIF
C
C  OTHERWISE, DETERMINE REQUIRED FIRST DAY OF AVAILABILITY
C
      IF (LAST.EQ.DR) THEN
         IBEG=1
      ELSE
         IBEG=LAST+1
      ENDIF
C
C  FIND AN INSPECTOR WITH POSITIVE SLACK WITH FIRST DAY OF AVAILABILITY
C  ON DAY IBEG
C
5     DO 10 K=1,II-1
C
C  FOR EACH TIME A NEW DEPOT BLOCK IS ENCOUNTERED, LOOK AHEAD TO
C  SEE IF ANOTHER INSPECTOR IS NEEDED TO MEET DEMAND AND GIVE MOST
C  RECENTLY ADDED INSPECTORS PRIORITY IN DAILY ASSIGNMENTS
C
         IF (IDEPOT.NE.DEP) THEN
            CALL LOOK
            IDEPOT=DEP
            I=II-K
         ELSE
            I=K
         ENDIF
         IF (SLKSUM(I).EQ.0) GOTO 10
         D1=DEMAND
         IF (SBEG(I).LE.IBEG.AND.SEND(I).GT.IBEG) THEN
            TA=SLKSUM(I)
            DAY=IBEG
            DA=SEND(I)-IBEG+1
            IF (DR-IBEG+1.GE.DA) THEN
               IDAYS=DA
```

```
C  SUFFICIENT SLACK SO ASSIGN INSPECTOR FOUND
C
               DEMAND=D1
               NI=SINSP(I)
               NEWINSP=0
               GOTO 50
            ENDIF
10       CONTINUE
C
C  TOTAL INSPECTORS IS 0 OR NO INSPECTOR FOUND WITH SUFFICIENT SLACK
C  SO GENERATE NEW INSPECTOR WITH FIRST DAY OF AVAILABILITY ON NEXTSV
C
20       CALL ADDINS
         GOTO 5
C
C  ASSIGN INSPECTOR TO A TOUR AND RETURN DEMAND AMOUNT SATISFIED
C
50       TFLAG=0
         CALL TOURS(NI,1,DEMAND,IDAYS,IBEG,IBEG+IDAYS-1)
         RETURN
         END
C
C
C
         SUBROUTINE LOCDAT(DEMAND)
C
C  RECORD INSPECTOR ASSIGNMENT AT LOCATION INODE WITH AMOUNT OF TIME
C  SPENT SPECIFIED BY DEMAND OVER PERIOD BETWEEN FIRST AND LAST
C
$INCLUDE: 'COMMONS.INS'
         LOC(KK)=INODE
         INSP(KK)=NI
         TBEG(KK)=FIRST
         TEND(KK)=LAST
         TIME(KK)=DEMAND
         FRQ(KK)=1
         ADNLTR(KK)=INNDTR
         ACDAYS(KK)=LAST-FIRST+1
         KK=KK+1
         RETURN
         END
```

```
      SUBROUTINE SLKDAT
C
C  SUMARIZE CONSTANT SLACK PERIODS FOR EACH INSPECTOR INTO SLACK
C  SUMMARY ARRAYS
C
$INCLUDE: 'COMMONS.INS'
      II=1
      DO 50 I=1,TI
         SCNT=1
         SFIRST=1
         DO 25 J=1,DR

C
C  DETERMINE CONSTANT SLACK PERIOD
C
            IF (J.EQ.DR.OR.SLACK(I,J).NE.SLACK(I,J+1)) THEN
               SINSP(II)=I
               SBEG(II)=SFIRST
               SEND(II)=J
               SLKSUM(II)=SLACK(I,J)
               SDAYS(II)=SCNT
               SFLAG(II)=0
               SFIRST=J+1
               SCNT=1
               II=II+1
            ELSE
               SCNT=SCNT+1
            ENDIF
25       CONTINUE
50    CONTINUE
      RETURN
      END
C
      SUBROUTINE LOOK
C
C  SUBROUTINE LOOK TO ADD AN ADDITIONAL INSPECTOR IF DEMAND REQUIRED FOR
C  NEW DEPOT BLOCK IS GREATER THAN TIME AVAILABLE OF CURRENT INSPECTORS
C
$INCLUDE: 'COMMONS.INS'
      DEMREQ=0
      TAVL=0
      DO 50 I=DEPLIST(NUMDEP),DEPLIST(NUMDEP+1)-1
         DO 25 J=1,P
            DEMREQ=DEMREQ+DEM(I,J)*FP(J)
25    CONTINUE
50    CONTINUE
      DO 100 I=1,II-1
         TAVL=TAVL+SLKSUM(I)*SDAYS(I)
100   CONTINUE
      IF (DEMREQ.GT.TAVL) CALL ADDINS
      RETURN
      END
```

```
****  PROGRAM MODULE ASSIGN2  ****
*
      SUBROUTINE ASSIGN2(J)
$INCLUDE: 'COMMONS.INS'
      TA=TAMAX
      DFIRM=0
C
C  INITIALIZE TEMPORARY STORAGE ARRAYS TO 0
C
      INTEGER DTOUR(MAXINSP)
      REAL DSAT(MAXINSP), DTRVL(MAXINSP), DADTRVL(MAXINSP)
5     DATA DTOUR/MAXINSP*0/
      DATA DSAT/MAXINSP*0./
      DATA DTRVL/MAXINSP*0./
      DATA DADTRVL/MAXINSP*0./
C
C  SET TFLAG AND AMOUNT OF DEMAND REQUIRED TO BE SATISFIED
C
      TFLAG=2
      DMIN=DEM(INODE,J)
C
C  PHASE 1:  DETERMINE IF INODE IS VISITED IN A HIGHER FREQUENCY TOUR
C
10    DO 25 I=1,INSPCNT
         IF (TOTUSED(I,J).EQ.0) GOTO 25
         D1=DEM(INODE,J)
         IDAYS=TOTUSED(I,J)
C
C  CALL TOURS TO DETERMINE IF INODE IS ALREADY VISITED DURING
C  INSPECTORS' AVAILABILITY PERIODS AND IF NOT, INCLUDE TRAVEL TIME
C  IN SLACK TIME CALCULATIONS
C
         CALL TOURS(AVLINSP(I),J,D1,IDAYS,SBEG(INDEX(I)),SEND(INDEX(I)))
         IF (INNDTR.EQ.0) THEN
            DTRVL(I)=TRVL
            DADTRVL(I)=ADTRVL
            DTOUR(I)=0
         ELSE
            DTRVL(I)=0
            DADTRVL(I)=0
            DTOUR(I)=INNDTR
         ENDIF
25    CONTINUE
C
C  PHASE 2:  DETERMINE IF EACH INSPECTOR HAS SUFFICIENT SLACK TO TRAVEL
C  AND WHAT AMOUNT OF DEMAND, DMIN, THEY CAN SATISFY FROM THEIR SLACK
C
      DO 50 I=1,INSPCNT
         IF (TOTUSED(I,J).EQ.0) GOTO 50
         T=TOTUSED(I,J)
         DD=DAYSAVL(I)
         TOTSLK=(DMIN+DTRVL(I)+DADTRVL(I))*T
         TRVSLK=(DTRVL(I)+DADTRVL(I))*T
         SLKAVL=SLKSUM(INDEX(I))*DD
```

```
C
C  IF SLACK AVAILABLE IS LESS THAN SLACK TIME USED TO TRAVEL, THEN
C  AMOUNT SATISFIED IS 0 AND SET FLAG SO INSPECTOR'S AVAILABILITY PERIOD
C  IS REMOVED FROM FUTURE CONSIDERATION
C
          IF (SLKAVL.LE.TRVSLK) THEN
             DSAT(I)=0.
             SFLAG(INDEX(I))=1
             WRITE (4,30) AVLINSP(I), INODE, J
30           FORMAT (/,18X,'** INSPECTOR ',I2,' INSUFFICIENT SLACK **',
     +                 /,10X,'SHORTFALL ATTEMPTING TO SATISFY NODE ',I2,
     +                 ' PERIOD ',I2,' DEMAND',/)
C
C  IF SLACK REQUIRED IS GREATER THAN INSPECTOR'S SLACK AVAILABLE, ADJUST
C  AMOUNT OF DEMAND SATISFIED TO USE ALL SLACK TIME FROM THE PERIOD
C
          ELSEIF (TOTSLK.GT.SLKAVL) THEN
             DSAT(I)=(SLKSUM(INDEX(I))*DD/T)-(DTRVL(I)+DADTRVL(I))
C
C  IF SLACK AVAILABLE IS GREATER THAN SLACK REQUIRED, INSPECTOR CAN
C  SATISFY ALL DEMAND GIVEN BY DMIN
C
          ELSE
             DSAT(I)=DMIN
          ENDIF
50     CONTINUE
C
C  FIND THE MINIMUM DEMAND THAT CAN BE SATISFIED OF ALL INSPECTORS
C
       DO 75 I=1,INSPCNT
          IF (TOTUSED(I,J).EQ.0) GOTO 75
          IF (DSAT(I).LT.DMIN) THEN
             DMIN=DSAT(I)
          ENDIF
75     CONTINUE
C
C  IF MINIMUM DEMAND IS 0, FIND ANOTHER SET OF INSPECTORS AND RETRY
C  PROCEDURE
C
       IF (DMIN.EQ.0) THEN
          CALL INSDAT
          GOTO 5
C
C  OTHERWISE SET TUSED TO DEMAND TO BE ASSIGNED
C
       ELSE
          TUSED=DMIN
       ENDIF
```

```
C
C   PHASE 3:  ASSIGN TUSED DEMAND TO EACH INSPECTOR OF THE SET OVER THEIR
C   AVAILABILITY PERIODS
C
      TFLAG=0
      DO 100 I=1,INSPCNT
         IF (TOTUSED(I,J).EQ.0) GOTO 100
         T=TOTUSED(I,J)
         IDAYS=TOTUSED(I,J)
C
C   CALCULATE SLACK USED PER DAY OF AVAILABILITY, SLKPD, FOR EACH
C   INSPECTOR AND THE TIME SPENT PER VISIT, TPV
         D=IDAYS
         DD=DAYSAVL(I)
         SLKPD=(TUSED+DTRVL(I)+DADTRVL(I))*T/DD
         TPV=TUSED
C
C   MAKE TOUR ASSIGNMENT FOR INSPECTORS VISITING A NODE FOR THE FIRST
C   TIME
C
         IF (DTOUR(I).EQ.0) THEN
            CALL TOURS(AVLINSP(I),J,TPV,IDAYS,SBEG(INDEX(I)),
     +                SEND(INDEX(I)))
            DTOUR(I)=INNDTR
         ENDIF
C
C   RECORD ASSIGNMENT DATA
C
         LOC(KK)=INODE
         INSP(KK)=AVLINSP(I)
         TIME(KK)=TPV
         ADNLTR(KK)=DTOUR(I)
         ACDAYS(KK)=IDAYS
         FRQ(KK)=J
C
C   ADJUST DAILY SLACK OVER AVAILABILITY PERIOD BY SLKPD
C
         SLKSUM(INDEX(I))=SLKSUM(INDEX(I))-SLKPD
         KK=KK+1
         DO 80 L=SBEG(INDEX(I)),SEND(INDEX(I))
            SLACK(AVLINSP(I),L)=SLACK(AVLINSP(I),L)-SLKPD
80       CONTINUE
100   CONTINUE
C
C   DECREMENT DEMAND FOR INODE DURING PERIOD J BY TUSED AND CONTINUE
C   PROCESS IF NOT 0
C
      DEM(INODE,J)=DEM(INODE,J)-TUSED
      IF (DEM(INODE,J).EQ.0) RETURN
      GOTO 5
      END
```

```
****   PROGRAM MODULE TOURS   ****
*
      SUBROUTINE TOURS(NINSP,J,TUSED,IDAYS,IDYBEG,IDYEND)
$INCLUDE: 'COMMONS.INS'
      INTEGER STORE(MAXTRS)
      REAL STTIME(MAXTRS)
C
C  INITIALIZE TOUR VARIABLES
C
      SAVE=0.
      ITOUR=0
      ITOUR1=0
      INNDTR=0
      MAXDAYS=IDYEND-IDYBEG+1
      IDEP=DEPOT(NINSP)
C
C  MAKE NO TOUR ASSIGNMENT FOR INSPECTORS ASSIGNED TO THEIR OWN DEPOT
C
      IF (INODE.EQ.DEP.AND.IDEP.EQ.DEP) THEN
         IF (TFLAG.EQ.2) THEN
            INNDTR=-1
            RETURN
C
C  FOR DAILY ASSIGNMENTS, ADJUST AMOUNT OF DEMAND IF IT IS GREATER
C  THAN THE INSPECTOR'S TIME AVAILABLE
C
         ELSEIF (J.EQ.1) THEN
            IF (TUSED.GT.TA) THEN
               TUSED=TA
               INNDTR=-1
               RETURN
            ELSE
               INNDTR=-1
               RETURN
            ENDIF
C
C  FOR NON-DAILY ASSIGNMENTS, RETURN MAKING NO ADJUSTMENT
C
         ELSE
            INNDTR=-1
            RETURN
      ENDIF
C
C  IF INSPECTOR HAS NO TOURS ASSIGNED, GENERAT  FIRST ROUND-TRIP TOUR
C
      IF (NTO   NINSP).EQ.0) GOTO 200
C
C  OTHERWISE, EXAMINE ALL FEASIBLE TOURS FOR ALREADY VISITED AND MAXIMUM
C  SAVINGS CRITERIA
C
      DO 100 I=1,NTOUR(NINSP)
         TEMPTL=0.
```

```
C
C  CHECK THAT DAILY COVERAGE OF TOUR IS COMPATIBLE WITH CURRENT
C  AVAILABILITY PERIOD
C
         IF (TRFRO(NINSP,I).LT.IDAYS.OR.IDYBEG.LT.TRBEG(NINSP,I).
     +       OR.IDYEND.GT.TREND(NINSP,I)) THEN
            GOTO 100
         ELSE
C
C  IF COMPATIBLE, CHECK TOUR TO DETERMINE IF INODE IS ALREADY VISITED
C  AND SATISFY DEMAND AVAILABLE ON THE TOUR, IF FOUND
C
            CALL SRCHP(NINSP,I,J,TUSED,IDAYS,MAXDAYS,ITOUR1)
            IF (TFLAG.EQ.2.AND.INNDTR.GT.0) RETURN
            IF (ITOUR1.GT.0) GOTO 450
C
C  IF NOT ALREADY VISITED BUT COMPATIBLE, DETERMINE WHERE MAXIMUM
C  SAVINGS IS ATTAINED IF INODE IS INSERTED BETWEEN ANY NODES OF TOUR
C
            CALL SAVMAX(NINSP,I,J,TUSED,IDAYS,MAXDAYS)
         ENDIF
100   CONTINUE
C
C  AFTER ALL TOURS CHECKED, IF SAVINGS FOUND, INSERT INODE INTO A TOUR
C
      IF (SAVE.GT.0) GOTO 250
C
C  OTHERWISE, GENERATE NEW ROUND-TRIP TOUR TO INODE FOR INSPECTOR,
C  DETERMINING TRAVEL TIME, AND AMOUNT OF DEMAND THAT CAN BE SATISFIED
C  WITHIN INSPECTOR'S AVAILABILITY CONSTRAINT FOR J=1
C
200   ADTRVL=0.
      TRVL=2*(TT(IDEP,INODE)/60.)
      IF (TUSED.GT.TA-TRVL.AND.J.EQ.1) THEN
         TUSED=TA-TRVL
      ENDIF
C
C  WHEN TFLAG > 0, RETURN TRAVEL INFORMATION BUT DON'T ASSIGN TOUR
C
      IF (TFLAG.EQ.1.OR.TFLAG.EQ.2) THEN
         INNDTR=0
         RETURN
      ENDIF
C
C  WHEN TFLAG = 0, MAKE TOUR ASSIGNMENT FOR NINSP
C
      NTOUR(NINSP)=NTOUR(NINSP)+1
      NODE1=IDEP
      NODE2=INODE
```

```
C
C  FIND SHORTEST TIME PATH BETWEEN INSPECTOR'S DEPOT AND INODE AND
C  FORM TOUR
C
      CALL FNDPTH(NODE1,NODE2)
      DO 210 M=1,NCNT-1
         MRET=2*NCNT-M
         TOUR(NTOUR(NINSP),M,NINSP)=SP(M)
         TOUR(NTOUR(NINSP),MRET,NINSP)=SP(M)
210   CONTINUE
C
C  STORE REMAINING TOUR INFORMATION AND CONSTRAINTS
C
      TOUR(NTOUR(NINSP),NCNT,NINSP)=SP(NCNT)
      TL(NINSP,NTOUR(NINSP))=TRVL+TUSED
      NNT(NINSP,NTOUR(NINSP))=2*NCNT-2
      TRFRQ(NINSP,NTOUR(NINSP))=IDAYS
      TRBEG(NINSP,NTOUR(NINSP))=IDYBEG
      TREND(NINSP,NTOUR(NINSP))=IDYEND
      TRLIM(NINSP,NTOUR(NINSP))=TA
      TSN(NTOUR(NINSP),NCNT,NINSP)=TUSED
      INNDTR=NTOUR(NINSP)
      RETURN
C
C  MAXIMUM SAVINGS ON TOUR ITOUR OF NINSP DETERMINED TO OCCUR WHEN
C  INODE IS INSERTED BETWEEN NODES INDEXED BY NODIND AND NODIND+1 SO
C  DETERMINE ADDITIONAL TRAVEL TIME REQUIRED AND AMOUNT OF DEMAND THAT
C  CAN BE SATISFIED WITHIN INSPECTOR'S AVAILABILITY CONSTRAINTS
C
250   NDCNT=0
      TRVL=0.
      ND1=TOUR(ITOUR,NODIND,NINSP)
      ND2=TOUR(ITOUR,NODIND+1,NINSP)
      ADTRVL=(TT(ND1,INODE)+TT(INODE,ND2)-TT(ND1,ND2))/60.
      T1=TUSED
      T2=TUSED
      IF (TUSED+ADTRVL.GT.TA.AND.J.EQ.1) THEN
         T1=TA-ADTRVL
         ENDIF
      ENDIF
      IF (SAVETL.GT.TRLIM(NINSP,ITOUR).AND.J.EQ.1) THEN
         T2=TUSED-(SAVETL-TRLIM(NINSP,ITOUR))
         ENDIF
      ENDIF
      IF (J.EQ.1) THEN
         TUSED=MIN(T1,T2)
      ELSE
         T=MIN(T1,T2)
         IF (T.EQ.TUSED) GOTO 260
         D=IDAYS
         IF (T.EQ.T1) THEN
            DD=ID1
            TUSED=TUSED*D/DD
            IDAYS=ID1
```

```
          ELSE
             DD=ID2
             TUSED=TUSED*D/DD
             IDAYS=ID2
          ENDIF
       ENDIF
C
C  IF TFLAG > 0, RETURN TRAVEL INFORMATION BUT DON'T ASSIGN TOUR
C
260    IF (TFLAG.EQ.1.OR.TFLAG.EQ.2) THEN
          INNDTR=0
          RETURN
       ENDIF
C
C  FORM NEW TOUR FROM THE ORIGINAL TOUR ITOUR WITH SHORTEST PATHS FROM
C  NODIND TO INODE AND INODE TO NODIND+1 INSERTED USING TEMPORARY
C  ARRAYS STORE FOR THE NODES AND STTIME FOR TIMES SPENT AT NODES
C
       DO 300 N=1,NODIND
          NDCNT=NDCNT+1
          STORE(NDCNT)=TOUR(ITOUR,N,NINSP)
          STTIME(NDCNT)=TSN(ITOUR,N,NINSP)
300    CONTINUE
       NODE1=TOUR(ITOUR,NODIND,NINSP)
       NODE2=INODE
       CALL FNDPTH(NODE1,NODE2)
       DO 325 N=2,NCNT
          NDCNT=NDCNT+1
          STORE(NDCNT)=SP(N)
C
C  SET TIME SPENT AT INODE TO TIME GIVEN BY TUSED
C
          IF (SP(N).EQ.INODE) THEN
             STTIME(NDCNT)=TUSED
          ELSE
             STTIME(NDCNT)=0.
          ENDIF
325    CONTINUE
       NODE1=INODE
       NODE2=TOUR(ITOUR,NODIND+1,NINSP)
       CALL FNDPTH(NODE1,NODE2)
       DO 350 N=2,NCNT
          NDCNT=NDCNT+1
          STORE(NDCNT)=SP(N)
          IF (SP(N).EQ.NODE2) THEN
             STTIME(NDCNT)=TSN(ITOUR,NODIND+1,NINSP)
          ELSE
             STTIME(NDCNT)=0.
          ENDIF
350    CONTINUE
       DO 375 N=NODIND+2,NNT(NINSP,ITOUR)+1
          NDCNT=NDCNT+1
          STORE(NDCNT)=TOUR(ITOUR,N,NINSP)
          STTIME(NDCNT)=TSN(ITOUR,N,NINSP)
```

```
375   CONTINUE
C
C  IF NUMBER OF DAYS REQUIRED TO VISIT INODE IS LESS THAN ORIGINAL
C  TOUR FREQUENCY, TOUR WITH INODE INSERTED REPRESENTS A NEW TOUR AND
C  ORIGINAL REMAINS UNCHANGED EXCEPT FOR ITS TOUR FREQUENCY, WHICH IS
C  REDUCED BY THE FREQUENCY OF NEW TOUR
C
      T=TRLIM(NINSP,ITOUR)
      TLSAVE=TL(NINSP,ITOUR)
      IF (TRFRQ(NINSP,ITOUR).GT.IDAYS) THEN
         TRFRQ(NINSP,ITOUR)=TRFRQ(NINSP,ITOUR)-IDAYS
         NTOUR(NINSP)=NTOUR(NINSP)+1
         ITOUR=NTOUR(NINSP)
         TRFRQ(NINSP,ITOUR)=IDAYS
         TRBEG(NINSP,ITOUR)=IDYBEG
         TREND(NINSP,ITOUR)=IDYEND
      ENDIF
C
C  STORE PERMANENT TOUR ASSIGNMENT FOR NINSP VISITING INODE
C
      DO 400 K=1,NDCNT
         TOUR(ITOUR,K,NINSP)=STORE(K)
         TSN(ITOUR,K,NINSP)=STTIME(K)
400   CONTINUE
C
C  STORE REMAINING TOUR INFORMATION
C
      NNT(NINSP,ITOUR)=NDCNT-1
      INNDTR=ITOUR
      TRLIM(NINSP,ITOUR)=T
      TL(NINSP,ITOUR)=TLSAVE+ADTRVL+TUSED
      RETURN
C
C  INODE FOUND TO BE VISITED ON TOUR ITOUR1 OF NINSP SO ADDITIONAL
C  TRAVEL TIME IS 0
C
C  IF TFLAG IS 1, RETURN NO ADDED TRAVEL TIME BUT DON'T ASSIGN
C
450   IF (TFLAG.EQ.1) THEN
         INNDTR=ITOUR1
         RETURN
      ENDIF
C
C  IF NUMBER OF DAYS REQUIRED TO VISIT INODE IS LESS THAN THE TOUR'S
C  FREQUENCY, FORM A NEW TOUR INCLUDING NEW VISITS TO INODE AND LEAVE
C  THE ORIGINAL TOUR UNCHANGED EXCEPT FOR ITS TOUR FREQUENCY WHICH IS
C  REDUCED BY THE NUMBER OF VISITS TO INODE
C
      IF (TRFRQ(NINSP,ITOUR1).GT.IDAYS) THEN
         TRFRQ(NINSP,ITOUR1)=TRFRQ(NINSP,ITOUR1)-IDAYS
         NTOUR(NINSP)=NTOUR(NINSP)+1
         NNT(NINSP,NTOUR(NINSP))=NNT(NINSP,ITOUR1)
         DO 460 K=1,NNT(NINSP,NTOUR(NINSP))+1
            TOUR(NTOUR(NINSP),K,NINSP)=TOUR(ITOUR1,K,NINSP)
```

```
      SUBROUTINE FNDPTH(NODE1,NODE2)
C
C  SUBROUTINE TO FIND THE SHORTEST TIME PATH BETWEEN NODES SPECIFIED
C  AS NODE1 AND NODE2 USING SHORTEST CHAIN MATRIX SC
C
$INCLUDE: 'COMMONS.INS'
      NCNT=1
      SP(NCNT)=NODE1
      NCNT=NCNT+1
10    SP(NCNT)=SC(NODE1,NODE2)
C
C  WHEN NODE2 IS NEXT INTERMEDIATE NODE, SHORTEST PATH COMPLETE
C
      IF (SP(NCNT).EQ.NODE2) RETURN
      NODE1=SP(NCNT)
      NCNT=NCNT+1
      GOTO 10
      END
C
C
      SUBROUTINE SAVMAX(NINSP,I,J,TUSED,IDAYS,MAXDAYS)
C
C  SUBROUTINE TO FIND MAXIMUM SAVINGS POSSIBLE IF INODE IS INSERTED
C  BETWEEN ANY TWO NODES OF TOUR I OF NINSP DURING FREQUENCY PERIOD J
C  TO SPEND TUSED AMOUNT OF TIME FOR NUMBER OF DAYS EQUAL TO IDAYS NOT
C  TO EXCEED MAXDAYS
C
$INCLUDE: 'COMMONS.INS'
      TOTTRV=0.
      ZERO=.01
C
C  IF TOUR I HAS NO TIME AVAILABLE, RETURN
C
      IF (TRLIM(NINSP,I)-TL(NINSP,I).LT.ZERO) RETURN
C
C  CHECK SAVINGS BETWEEN EACH PAIR OF NODES DENOTED ITEMP1 AND ITEMP2
C
      DO 50 K=1,NNT(NINSP,I)
         ITEMP1=TOUR(I,K,NINSP)
         ITEMP2=TOUR(I,K+1,NINSP)
         IF (ITEMP1.EQ.INODE.OR.ITEMP2.EQ.INODE) GOTO 50
         S=SAVING(NINSP,TOTTRV,ITEMP1,ITEMP2)
         IF (S.EQ.0) GOTO 50
C
C  IF SAVINGS DETERMINED IS GREATER THAN CURRENT MAXIMUM SAVINGS STORED
C  AS SAVE, CHECK THAT SOME AMOUNT OF DEMAND CAN BE SATISFIED
C
         IF (S.GT.SAVE) THEN
            ADTRVL=TT(ITEMP1,INODE)+TT(INODE,ITEMP2)-TT(ITEMP1,ITEMP2)
            ADTRVL=ADTRVL/60.
```

```
C  IF ADDED TRAVEL TIME IS GREATER THAN INSPECTOR'S SLACK TIME OR
C  AMOUNT OF TIME AVAILABLE ON THE TOUR, DON'T MAKE INSERTION
C
            IF (ADTRVL.GE.TA.OR.ADTRVL.GE.TRLIM(NINSP,I)-TL(NINSP,I))
     +         GOTO 50
C
C  IF FIRM AMOUNT OF DEMAND IS REQUIRED AND PROPOSED NEW TOUR LENGTH
C  EXCEEDS INSPECTOR'S TIME AVAILABLE OR MAXIMUM TOUR LIMIT, DON'T
C  CONSIDER INSERTION
C
            TEMPTL=ADTRVL+TUSED+TL(NINSP,I)
            IF ((TEMPTL.GT.TA.OR.TEMPTL.GT.TRLIM(NINSP,I)).AND.
     +          DFIRM.EQ.1) GOTO 50
C
C  IF PROPOSED NEW TOUR LENGTH IS GREATER THAN INSPECTOR'S TIME
C  AVAILABLE FOR NON-DAILY FREQUENCY PERIODS, DETERMINE NUMBER OF DAYS
C  REQUIRED TO MEET DEMAND
C
            IF (TEMPTL.GT.TA.AND.J.GE.2) THEN
               T=TA-ADTRVL
               ID=CALCDAY(IDAYS,TUSED,T)
C
C  IF NUMBER OF DAYS TO MEET DEMAND IS GREATER THAN SLACK AVAILABILITY
C  PERIOD, DON'T MAKE INSERTION
C
               IF (ID.GT.TRFRQ(NINSP,I)) GOTO 50
            ENDIF
C
C  IF PROPOSED NEW TOUR LENGTH IS GREATER THAN THE TOUR LIMITING TIME
C  FOR NON-DAILY FREQUENCY PERIODS, DETERMINE NUMBER OF DAYS REQUIRED
C  TO MEET DEMAND
C
            IF (TEMPTL.GT.TRLIM(NINSP,I).AND.J.GE.2) THEN
               T=TUSED-(TEMPTL-TRLIM(NINSP,I))
               ID=CALCDAY(IDAYS,TUSED,T)
C
C  IF NUMBER OF DAYS TO MEET DEMAND IS GREATER THAN SLACK AVAILABILITY
C  PERIOD, DON'T MAKE INSERTION
C
               IF (ID.GT.TRFRQ(NINSP,I)) GOTO 50
            ENDIF
C
C  OTHERWISE, STORE NEW MAXIMUM SAVING AS SAVE, THE TOUR NUMBER AS
C  ITOUR, PROPOSED TOUR LENGTH AS SAVETL, AND THE NODE INDEX AFTER WHICH
C  INODE IS TO BE INSERTED AS NODIND
C
            SAVE=S
            ITOUR=I
            SAVETL=TEMPTL
            NODIND=K
         ENDIF
50    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SRCHP(NINSP,I,J,TUSED,IDAYS,MAXDAYS,ITOUR1)
C
C  SUBROUTINE TO FIND IF NODE INODE REQUIRING TUSED DEMAND DURING
C  FREQUENCY PERIOD J FOR NUMBER OF DAYS SPECIFIED BY IDAYS,NOT TO
C  EXCEED MAXDAYS IS ALREADY VISITED IN TOUR I OF NINSP
C
$INCLUDE: 'COMMONS.INS'
      ZERO=.01
      T1=TUSED
      T2=TUSED
C
C  CHECK EVERY NODE OF TOUR I, NOT INCLUDING DEPOTS, FOR OCCURANCE
C  OF INODE
C
      DO 50 K=2,NNT(NINSP,I)
         ITEMP=TOUR(I,K,NINSP)
C
C  INODE FOUND WITH SOME TIME SPENT AND TFLAG SET AT 2 SO RETURN TOUR
C  NUMBER TO ALLOW AS MUCH SLACK TIME AS POSSIBLE TO BE SATISFIED ON
C  THIS TOUR
C
         IF (ITEMP.EQ.INODE.AND.TSN(I,K,NINSP).GT.0.AND.TFLAG.EQ.2) THEN
            INNDTR=I
            RETURN
         ENDIF
         IF (ITEMP.NE.INODE) GOTO 50
C
C  OCCURANCE OF INODE FOUND, THEN ENSURE TOUR CAPABLE OF SATISFYING
C  MORE DEMAND
C
         IF (TRLIM(NINSP,I)-TL(NINSP,I).LT.ZERO) GOTO 50
C
C  IF DEMAND IS A FIRM REQUIREMENT AND IT IS GREATER THAN INSPECTOR'S
C  TIME AVAILABLE OR THE AMOUNT OF TIME LEFT ON THE TOUR, DON'T SATISFY
C  THE DEMAND OF INODE ON THIS TOUR
C
         IF ((TUSED.GT.TA.OR.TUSED+TL(NINSP,I).GT.TRLIM(NINSP,I)).AND.
     +         DFIRM.EQ.1) GOTO 50
C
C  IF DEMAND IS GREATER THAN INSPECTOR'S TIME AVAILABLE, DETERMINE
C  NUMBER OF DAYS REQUIRED TO SATISFY DEMAND
C
         IF (TUSED.GT.TA) THEN
            T1=TA
            IF (J.GE.2) THEN
               ID1=CALCDAY(IDAYS,TUSED,T1)
C
C  IF REQUIRED NUMBER OF DAYS TO MEET DEMAND IS GREATER THAN SLACK
C  AVAILABILITY PERIOD, DON'T SATISFY DEMAND ON THIS TOUR
C
               IF (ID1.GT.TRFRQ(NINSP,I)) GOTO 50
            ENDIF
         ENDIF
```

```
C  IF DEMAND IS GREATER THAN AMOUNT OF TIME AVAILABLE ON THE TOUR,
C  DETERMINE NUMBER OF DAYS REQUIRED TO SATISFY DEMAND
C
          IF (TUSED+TL(NINSP,I).GT.TRLIM(NINSP,I)) THEN
             T2=TRLIM(NINSP,I)-TL(NINSP,I)
             IF (J.GE.2) THEN
                ID2=CALCDAY(IDAYS,TUSED,T2)
C
C  IF REQUIRED NUMBER OF DAYS TO MEET DEMAND IS GREATER THAN SLACK
C  AVAILABILITY PERIOD, DON'T SATISFY DEMAND ON THIS TOUR
C
                IF (ID2.GT.TRFRQ(NINSP,I)) GOTO 50
             ENDIF
          ENDIF
C
C  FOR DAILY ASSIGNMENTS, DETERMINE AMOUNT OF DEMAND THAT CAN BE
C  SATISFIED AS MINIMUM OF INSPECTOR TIME AVAILABLE, TIME AVAILABLE ON
C  TOUR, AND TOTAL DEMAND TO BE SATISFIED, TUSED
C
          IF (J.EQ.1) THEN
             TUSED=MIN(T1,T2)
          ENDIF
C
C  RETURN TOUR FOUND WITH INODE, ITOUR1, ITS TOUR LENGTH, AND THE NODE
C  INDEX FOR INODE, NODIND
C
25        ITOUR1=I
          SAVETL=TL(NINSP,I)
          NODIND=K
          RETURN
50     CONTINUE
C
C  RETURN WITH NO TOUR FOUND WITH INODE CAPABLE OF SATISFYING DEMAND
C
       RETURN
       END
C
       FUNCTION CALCDAY(IDAYS,TUSED,T)
C
C  FUNCTION TO CALCULATE THE NUMBER OF DAYS REQUIRED TO MEET DEMAND OF
C  TUSED OVER IDAYS NUMBER OF DAYS WHEN ONLY AN AMOUNT OF TIME T IS
C  ALLOWABLE PER DAY
C
$INCLUDE: 'COMMONS.INS'
       D=IDAYS
       DD=D*TUSED/T
C
C  ROUND ID TO NEXT HIGHER INTEGER
       ID=NINT(DD)
       IF (ID.GE.DD) GOTO 10
       ID=ID+1
10     CALCDAY=ID
       RETURN
       END
```

```
****  PROGRAM MODULE OUTPUT  ****
*
      SUBROUTINE TROUT
C
C  SUBROUTINE FOR TOUR OUTPUT
C
$INCLUDE: 'COMMONS.INS'
      WRITE (4,50)
      DO 30 K=1,TI
         WRITE (4,51) K
         DO 20 I=1,NTOUR(K)
            IFOUND=0
            WRITE (4,52) I
            DO 10 J=1,NNT(K,I)+1
               WRITE (4,53) TOUR(I,J,K), TSN(I,J,K)
10          CONTINUE
            WRITE (4,54) TL(K,I), TRLIM(K,I)
            WRITE (4,55) TRBEG(K,I), TREND(K,I), TRFRQ(K,I)
            WRITE (4,60)
            WRITE (4,61)
            DO 15 L=1,KK-1
               IF (INSP(L).EQ.K.AND.ADNLTR(L).EQ.I.AND.ACDAYS(L).
     +             LT.TRFRQ(K,I)) THEN
                  IFOUND=1
                  WRITE (4,62) LOC(L), TIME(L), ACDAYS(L),
     +                            FRQ(L)
               ENDIF
15          CONTINUE
            IF (IFOUND.EQ.0) THEN
               WRITE (4,63)
            ENDIF
20       CONTINUE
30    CONTINUE
50    FORMAT (//,1X,'************ SUMMARY OF TOURS ************')
51    FORMAT (//,' ***** TOURS OF INSPECTOR ',I2,' *****')
52    FORMAT (/,4X,'TOUR NUMBER ',I2,/)
53    FORMAT (12X,'NODE ',I2,5X,'TIME SPENT ',F6.2,' HOURS')
54    FORMAT (/,8X,'TOUR LENGTH: ',F5.2,'    MAXIMUM TIME OF TOUR:  ',
     +        F5.2)
55    FORMAT (8X,'SLACK DAYS WHEN TOUR POSSIBLE:  ',I3,' TO ',I3,
     +        '     DAYS REQUIRED: ',I3)
60    FORMAT (/,8X,'TOUR UTILIZATION FROM SLACK TIME:')
61    FORMAT (/,12X,'NODE   TIME SPENT   DAYS   FREQ PERIOD',/)
62    FORMAT (13X,I2,6X,F6.2,6X,I3,9X,I2)
63    FORMAT (20X,'--NONE--')
      RETURN
      END
```

```
      SUBROUTINE IDATOUT
C
C  SUBROUTINE FOR OUTPUT OF INSPECTOR DATA AS PRODUCED IN INSDAT MODULE
C
$INCLUDE: 'COMMONS.INS'
      WRITE (4,110)
      DO 50 J=1,INSPCNT
         WRITE (4,115) J, AVLINSP(J), DAYSAVL(J), HRSAVL(J), INDEX(J)
50    CONTINUE
      WRITE (4,120)
      DO 100 I=1,INSPCNT
         WRITE (4,130) AVLINSP(I), (TOTUSED(I,J), J=1,P)
100   CONTINUE
110   FORMAT (///,5X,' ******* INSPECTORS WITH SLACK OVER TOTAL',
     +         ' WORKING PERIOD *******',//,10X,'I    INSP    DAYS',
     +         ' AVAIL    HRS AVAIL    INDEX',/)
115   FORMAT (9X,I2,4X,I2,8X,I3,8X,F6.2,6X,I2)
120   FORMAT (//,10X,'EQUIVALENT AVAILABILITIES IN DAYS:',//,
     +    '                            FREQUENCY CLASS',//,5X,
     +    '  INSP    1    2    3    4    5    6    7    8    9   10',/)
130   FORMAT (8X,I2,3X,10(I3,2X))
      RETURN
      END
C
C
C
      SUBROUTINE SLKOUT
C
C  SUBROUTINE FOR OUTPUT OF SLACK SUMMARY ARRAYS
C
$INCLUDE: 'COMMONS.INS'
      WRITE (4,55)
      DO 50 L=1,II-1
         WRITE (4,56) L, SINSP(L), SBEG(L), SEND(L), SLKSUM(L), SDAYS(L)
50    CONTINUE
55    FORMAT (/,5X,'(II)',3X,'INSP',3X,'BEG',3X,'END',3X,'SLACK',
     +         3X,'DAYS',/)
56    FORMAT (6X,I2,5X,I2,4X,I3,3X,I3,2X,F6.2,4X,I3)
      RETURN
      END
C
C
C
      SUBROUTINE ADOUT
C
C  SUBROUTINE FOR THE OUTPUT OF ASSIGNMENT DATA FOR ALL NODES
C
$INCLUDE: 'COMMONS.INS'
      TDEM=0.
      WRITE (4,110)
      DO 100 N=1,NN
         WRITE (4,115) N
         WRITE (4,120)
         DO 80 J=1,P
```

```
            TDEM=0.
            DO 50 I=1,KK-1
               IF (LOC(I).EQ.N.AND.FRQ(I).EQ.J) THEN
                  DEMSAT=TIME(I)*ACDAYS(I)
                  TDEM=TDEM+DEMSAT
                  IF (ADNLTR(I).EQ.-1) THEN
                     WRITE (4,124) J, INSP(I), TIME(I), ACDAYS(I),
     +                             DEMSAT
                  ELSE
                     WRITE (4,125) J, INSP(I), ADNLTR(I), TIME(I),
     +                             ACDAYS(I), DEMSAT
                  ENDIF
               ENDIF
50          CONTINUE
            IF (TDEM.EQ.0) GOTO 80
            WRITE (4,130) TDEM
80       CONTINUE
100   CONTINUE
110   FORMAT (///,' **** SUMMARY OF DEMAND SATISFACTION BY INSPECTOR ',
     +        '****',/)
115   FORMAT (///,3X,'NODE ',I2,':',/)
120   FORMAT (3X,'FRQ    INSP    REF TOUR    TIME PER VIS     ',
     +        'DAYS      DEM HRS',/)
124   FORMAT (3X,I2,5X,I2,6X,'DEP',9X,F6.2,8X,I3,7X,F7.2)
125   FORMAT (3X,I2,5X,I2,7X,I2,9X,F6.2,8X,I3,7X,F7.2)
130   FORMAT (45X,'TOTAL:   ',F7.2,/)
      RETURN
      END
C
C
C
      SUBROUTINE LOCOUT
C
C  SUBROUTINE TO OUTPUT MOST RECENT ASSIGNMENT INFORMATION
C
$INCLUDE: 'COMMONS.INS'
      WRITE (4,10)
10    FORMAT (/,5X,'LOC    INSP    FIRST DAY    LAST DAY    TIME    ',
     +        'TOTAL DAYS    DEM HRS',/)
      DO 20 I=KKCNT,KK-1
         DEMSAT=ACDAYS(I)*TIME(I)
         IF (TBEG(I).EQ.0.AND.TEND(I).EQ.0) THEN
            WRITE (4,24) LOC(I), INSP(I), TIME(I), ACDAYS(I), DEMSAT
         ELSE
            WRITE (4,25) LOC(I), INSP(I), TBEG(I), TEND(I), TIME (I),
     +                   ACDAYS(I), DEMSAT
         ENDIF
20    CONTINUE
24    FORMAT (6X,I2,4X,I2,4X,'.....FROM SLACK.....',2X,F6.2,7X,I3,
     +        5X,F7.2)
25    FORMAT (6X,I2,4X,I2,7X,I3,9X,I3,4X,F6.2,7X,I3,5X,F7.2)
      RETURN
      END
```

```
      SUBROUTINE STATS
C
C  SUBROUTINE TO PROVIDE SEPARATE SUMMARY OF INPUT DATA AND
C  ALGORITHM PERFORMANCE STATISTICS
C
$INCLUDE: 'COMMONS.INS'
C
C  OUTPUT INPUT PARAMETERS
C
      WRITE (5,10)
      WRITE (5,15) TDA, DR, TAMAX
      WRITE (5,20) NN
      DO 5 I=1,NUMDEP
         WRITE (5,25) I, DEPLIST(I)
5     CONTINUE
10    FORMAT (//,29X,'SUMMARY OF INPUT DATA',//)
15    FORMAT (10X,'TOTAL ANNUAL WORKING DAYS PER INSPECTOR:  ',I3,/,10X,
     +         'TOTAL ANNUAL WORK DAYS REQUIRED:  ',I3,/,10X,
     +         'INSPECTOR HOURS AVAILABLE PER DAY:  ',F6.2,/)
20    FORMAT (10X,'TOTAL NUMBER OF NODES IN NETWORK:  ',I2)
25    FORMAT (15X,'DEPOT ',I2,' AT NODE ',I2)
C
C  OUTPUT FREQUENCY PERIODS USED
C
30    WRITE (5,50) P-1
      DO 40 J=2,P
         WRITE (5,55) J-1, FP(J)
40    CONTINUE
50    FORMAT (/,10X,'NUMBER OF NON-DAILY FREQUENCY PERIODS:  ',I2)
55    FORMAT (15X,'PERIOD ',I2,' -- FREQUENCY: ',I3,' PER YEAR')
C
C  OUTPUT SUMMARY OF DEMAND LOAD
C
      REWIND 2
      READ (2,*) ((DEM(I,J), J=1,P), I=1,NN)
      WRITE (5,60)
60    FORMAT (//,27X,'SUMMARY OF DEMAND LOADING',/)
      DO 100 J=1,P
         WRITE (5,110) FP(J)
         DO 80 I=1,NN
            FD=FD+DEM(I,J)*FP(J)
80       CONTINUE
         WRITE (5,115) FD
         IF (J.GE.2) THEN
            CD=CD+FD
         ENDIF
         TD=TD+FD
         FD=0.
100   CONTINUE
      DD=TD-CD
      WRITE (5,120) DD, CD, TD
110   FORMAT (10X,'ANNUAL FREQUENCY OF ',I3,':')
115   FORMAT (20X,'TOTAL DEMAND = ',F8.2,' HOURS PER YEAR')
120   FORMAT (/,10X,'TOTAL ANNUAL DAILY DEMAND =      ',F9.2,' HOURS',
```

```
     +          /,10X,'TOTAL ANNUAL NON-DAILY DEMAND =  ',F9.2,' HCURS',
     +          /,10X,'GRAND TOTAL OF ANNUAL DEMAND =   ',F9.2,' HOURS')
C
C  CALCULATE LOWER BOUND ESTIMATE OF MINIMUM NUMBER OF INSPECTORS
C
      WRITE (5,130)
      THRS=TDA*TAMAX
      XMIN=TD/THRS
      NI=NINT(XMIN)
      IF (NI.GE.XMIN) GOTO 125
      NI=NI+1
125   WRITE (5,135) NI, THRS
130   FORMAT (//,20X,'SUMMARY OF INSPECTOR UTILIZATION',//)
135   FORMAT (10X,'LOWER BOUND ON MINIMUM NUMBER OF INSPECTORS:   ',
     +         I2,/,10X,'ANNUAL INSPECTOR AVAILABILITY IN HOURS:   ',
     +         F8.2,/)
C
C  OUTPUT TOTAL NUMBER OF INSPECTOR ASSIGNED
C
      WRITE (5,140) TI
140   FORMAT (10X,'**** TOTAL OF ',I2,' INSPECTORS ASSIGNED  ****',/)
C
C  CALCULATE INSPECTOR UTILIZATION BY DETERMINING SLACK TIME REMAINING
C
      DO 200 I=1,TI
         DO 150 J=1,DR
            SLHRS=SLHRS+SLACK(I,J)
150      CONTINUE
         SUTIL=((THRS-SLHRS)/THRS)*100.
         TUTIL=TUTIL+SUTIL
         DO 180 N=1,KK-1
            IF (INSP(N).EQ.I) THEN
               TSINSP=TSINSP+(TIME(N)*ACDAYS(N))
            ENDIF
180      CONTINUE
C
C  OUTPUT INDIVIDUAL INSPECTOR UTILIZATION STATISTICS
C
         TIUTIL=(TSINSP/THRS)*100.
         TOTTI=TOTTI+TIUTIL
         TSTRVL=THRS-TSINSP-SLHRS
         TRUTIL=(TSTRVL/THRS)*100.
         TOTTR=TOTTR+TRUTIL
         WRITE (5,215) I, DEPOT(I), TSINSP, TIUTIL, TSTRVL, TRUTIL,
     +                  SLHRS, SUTIL
         SLHRS=0.
         TSINSP=0.
         TSTRVL=0.
200   CONTINUE
```

```
C  OUTPUT AVERAGE UTILIZATION STATISTICS
C
      AVUTIL=TUTIL/TI
      AVINSP=TOTTI/TI
      AVTRVL=TOTTR/TI
      WRITE (5,219)
      WRITE (5,220) AVINSP, AVTRVL, AVUTIL
215   FORMAT (10X,'INSPECTOR ',I2,' BASED AT NODE ',I2,':',/,15X,
     +        'TOTAL TIME INSPECTING = ',F7.2,'    UTILIZATION = ',
     +        F6.2,'%',/,15X,'TOTAL TIME TRAVELLING = ',F7.2,
     +        '    UTILIZATION = ',F6.2,'%',/,15X,
     +        'SLACK HOURS REMAINING = ',F7.2,'    UTILIZATION = ',
     +        F6.2,'%',/)
219   FORMAT (/,10X,'TOTALS FOR ALL INSPECTORS:')
220   FORMAT (15X,'AVERAGE TIME SPENT INSPECTING = ',F6.2,'%',/,
     +        15X,'AVERAGE TIME SPENT TRAVELLING = ',F6.2,'%',/,
     +        15X,'AVERAGE SLACK TIME UTILIZED =   ',F6.2,'%')
      WRITE (5,260)
C
C  OUTPUT NUMBER OF INSPECTORS ASSIGNED TO EACH DEPOT
C
      DO 300 N=1,NUMDEP
         DEP=DEPLIST(N)
         DO 250 I=1,TI
            IF (DEP.EQ.DEPOT(I)) THEN
               ICNT=ICNT+1
            ENDIF
250      CONTINUE
         WRITE (5,270) DEP, ICNT
         ICNT=0
300   CONTINUE
260   FORMAT (//,10X,'NUMBER OF INSPECTORS ASSIGNED TO EACH DEPOT')
270   FORMAT (15X,'DEPOT AT NODE ',I2,':  ',I2,' INSPECTORS')
      RETURN
      END
```

```
C  LISTING FOR 'COMMONS.INS'
C
C  INCLUDES VARIABLE ARRAY PARAMETERS AND DEFINITIONS OF GLOBAL ARRAYS
C  AND VARIABLES
C
      PARAMETER (MAXINSP=40,MAXNUM=200,MAXTIME=255,MAXNODE=20,MAXFRQ=7,
     +           MAXTRS=25, MAXAD=1000, MAXDEP=5)
      INTEGER LOC(MAXAD), INSP(MAXAD), TBEG(MAXAD), TEND(MAXAD),
     +        SINSP(MAXNUM), SBEG(MAXNUM), SEND(MAXNUM), SDAYS(MAXNUM),
     +        NV(MAXNODE), AVLINSP(MAXINSP), DAYSAVL(MAXINSP),
     +        INDEX(MAXINSP), FP(MAXFRQ), ACDAYS(MAXAD),
     +        TOTUSED(MAXINSP,MAXFRQ), REMAIN(MAXFRQ), SP(MAXNODE),
     +        NNT(MAXINSP,MAXTRS), TRFRQ(MAXINSP,MAXTRS),
     +        SC(MAXNODE,MAXNODE), TOUR(MAXTRS,MAXNODE,MAXINSP),
     +        NTOUR(MAXINSP), DEPOT(MAXINSP), SFLAG(MAXNUM),
     +        TRBEG(MAXINSP,MAXTRS), TREND(MAXINSP,MAXTRS),
     +        FRQ(MAXAD), ADNLTR(MAXAD), DEPLIST(MAXDEP), TFLAG,
     +        INNDTR, INSPCNT, IADD, DEP, NUMDEP, KKCNT, DFIRM,
     +        NNV, NI, DA, DR, P, NN, DAY, FIRST, LAST, INODE,
     +        KK, II, TI, TDA, NEXTSV
      REAL TIME(MAXAD), SLACK(MAXINSP,MAXTIME),
     +        SLKSUM(MAXNUM), DEM(MAXNODE,MAXFRQ), HRSAVL(MAXINSP),
     +        TT(MAXNODE,MAXNODE), TL(MAXINSP,MAXTRS),
     +        TSN(MAXTRS,MAXNODE,MAXINSP), TRLIM(MAXINSP,MAXTRS),
     +        ADTRVL, TRVL, TAMAX
      COMMON /LDAT/LOC, INSP, TBEG, TEND, NV, FP, DEPOT, DEPLIST
      COMMON /IDAT/SINSP, SBEG, SEND, SDAYS, AVLINSP, DAYSAVL,
     +              INDEX, TOTUSED, REMAIN, FRQ, ADNLTR, SFLAG
      COMMON /TDAT/TIME, SLACK, SLKSUM, DEM, HRSAVL, TT,
     +              ACDAYS, EQDAYS
      COMMON /TRDAT/NNT, TRFRQ, NTOUR, SC, TOUR, TRLIM,
     +               SP, TRBEG, TREND
      COMMON /DDAT/ TL, TSN
      COMMON /VARS/NI, DA, DR, P, NN, DAY, FIRST, LAST, INODE, KK,
     +              TA, TDD, II, SCNT, TI, TDA, NNV, DFIRM, INNDTR,
     +              DEP, INSPCNT, TRVL, ADTRVL, NCNT, TFLAG, NEXTSV,
     +              SAVE, ITOUR, NODIND, SAVETL, NUMDEP, TAMAX, KKCNT,
```

APPENDIX C: INPUT DATA SETS

DEMAND DATA SET 1

HOURS/DAY BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	8.30	.00	.00	8.70	6.00	.90	4.80
2	.00	.00	.00	1.20	.00	.00	.70
3	.00	.00	.00	.60	.00	.00	2.30
4	.00	.80	.00	2.60	.00	2.10	.40
5	.00	.00	.00	.00	.00	.00	.00
6	1.60	.50	.00	4.60	3.20	3.90	1.40
7	.00	.00	.00	2.60	.00	.00	.30
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.60	.00	.80	.00	.00	2.40
12	.00	.00	.00	.00	.00	.00	.40
13	.00	.00	.00	1.50	.00	.00	.60
14	2.00	.70	.00	7.70	.00	1.20	4.90
15	.00	.00	.00	.60	.00	.00	.00
16	.00	.60	.00	1.40	.00	.00	.00
17	.00	.80	.00	13.80	.00	.00	1.00
18	.00	.00	.00	1.00	.00	.00	.60
19	.80	2.50	.00	15.20	1.20	.00	8.40
TOTAL:	12.70	6.50	.00	62.30	10.40	8.10	28.20

TOTAL HOURS/YEAR BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	2091.60	.00	.00	104.40	24.00	1.80	4.80
2	.00	.00	.00	14.40	.00	.00	.70
3	.00	.00	.00	7.20	.00	.00	2.30
4	.00	41.60	.00	31.20	.00	4.20	.40
5	.00	.00	.00	.00	.00	.00	.00
6	403.20	26.00	.00	55.20	12.80	7.80	1.40
7	.00	.00	.00	31.20	.00	.00	.30
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	31.20	.00	9.60	.00	.00	2.40
12	.00	.00	.00	.00	.00	.00	.40
13	.00	.00	.00	18.00	.00	.00	.60
14	504.00	36.40	.00	92.40	.00	2.40	4.90
15	.00	.00	.00	7.20	.00	.00	.00
16	.00	31.20	.00	16.80	.00	.00	.00
17	.00	41.60	.00	165.60	.00	.00	1.00
18	.00	.00	.00	12.00	.00	.00	.60
19	201.60	130.00	.00	182.40	4.80	.00	8.40
TOTAL:	3200.40	338.00	.00	747.60	41.60	16.20	28.20

DEMAND DATA SET 2

HOURS/DAY BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	2.10	.00	.00	31.50	.20	.00	10.00
2	.00	.00	.00	.40	.00	.00	.90
3	.00	.00	.00	1.20	.40	1.20	8.60
4	.00	.40	.00	2.10	.00	5.50	9.10
5	.00	.00	.00	.00	.00	.00	.00
6	.00	.50	.00	20.40	5.40	6.20	18.00
7	.00	.40	.00	2.50	.00	.00	.50
8	.00	.00	.00	1.50	.00	.00	1.40
9	.00	.00	.00	1.00	.00	1.20	11.00
10	.00	.00	.00	.50	.00	.00	9.20
11	.00	.00	.00	.00	.00	1.20	8.50
12	.00	.00	.00	3.40	.00	.00	.40
13	.00	1.70	.00	3.30	.00	.00	1.00
14	.40	.60	.00	20.90	.00	.00	.00
15	.00	.00	.00	.00	.00	.00	2.30
16	.00	.40	.00	4.60	.00	.00	.20
17	.00	.00	.00	.00	.00	.00	.80
18	.00	.00	.00	.40	.00	.00	.00
19	.00	6.40	.00	92.30	1.00	1.20	25.30
TOTAL:	2.50	10.40	.00	186.00	7.00	16.50	107.20

TOTAL HOURS/YEAR BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	529.20	.00	.00	378.00	.80	.00	10.00
2	.00	.00	.00	4.80	.00	.00	.90
3	.00	.00	.00	14.40	1.60	2.40	8.60
4	.00	20.80	.00	25.20	.00	11.00	9.10
5	.00	.00	.00	.00	.00	.00	.00
6	.00	26.00	.00	244.80	21.60	12.40	18.00
7	.00	20.80	.00	30.00	.00	.00	.50
8	.00	.00	.00	18.00	.00	.00	1.40
9	.00	.00	.00	12.00	.00	2.40	11.00
10	.00	.00	.00	6.00	.00	.00	9.20
11	.00	.00	.00	.00	.00	2.40	8.50
12	.00	.00	.00	40.80	.00	.00	.40
13	.00	88.40	.00	39.60	.00	.00	1.00
14	100.80	31.20	.00	250.80	.00	.00	.00
15	.00	.00	.00	.00	.00	.00	2.30
16	.00	20.80	.00	55.20	.00	.00	.20
17	.00	.00	.00	.00	.00	.00	.80
18	.00	.00	.00	4.80	.00	.00	.00
19	.00	332.80	.00	1107.60	4.00	2.40	25.30
TOTAL:	630.00	540.80	.00	2232.00	28.00	33.00	107.20

DATA SET 3

HOURS/DAY BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	.50	3.40	.00	58.40	.00	.00	.00
2	.00	.00	.00	.00	.00	.00	.00
3	.00	.00	.00	2.50	.40	.00	.00
4	.00	3.00	.00	16.00	.00	.00	2.70
5	.00	.00	.00	.00	.00	.00	.00
6	.00	3.20	.00	13.30	3.30	.00	.00
7	.00	.00	.00	.30	.00	.00	.00
8	.00	.00	.00	2.40	.00	.00	8.20
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	2.70	.40	.00	3.30
12	.00	.00	.00	1.10	.00	.00	.00
13	.00	.00	.00	.00	.00	.00	.30
14	.00	.60	.00	38.90	.00	.00	.00
15	.00	.00	.00	1.10	1.10	1.20	.00
16	.00	.00	.00	9.50	.00	.00	.00
17	.00	.00	.00	15.60	.00	.00	.90
18	.00	.00	.00	.00	.00	.00	.00
19	.00	30.00	.00	13.70	2.20	.00	.00
TOTAL:	.50	40.20	.00	175.50	7.40	1.20	15.40

TOTAL HOURS/YEAR BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	126.00	176.80	.00	700.80	.00	.00	.00
2	.00	.00	.00	.00	.00	.00	.00
3	.00	.00	.00	30.00	1.60	.00	.00
4	.00	156.00	.00	192.00	.00	.00	2.70
5	.00	.00	.00	.00	.00	.00	.00
6	.00	166.40	.00	159.60	13.20	.00	.00
7	.00	.00	.00	3.60	.00	.00	.00
8	.00	.00	.00	28.80	.00	.00	8.20
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	32.40	1.60	.00	3.30
12	.00	.00	.00	13.20	.00	.00	.00
13	.00	.00	.00	.00	.00	.00	.30
14	.00	31.20	.00	466.80	.00	.00	.00
15	.00	.00	.00	13.20	4.40	2.40	.00
16	.00	.00	.00	114.00	.00	.00	.00
17	.00	.00	.00	187.20	.00	.00	.90
18	.00	.00	.00	.00	.00	.00	.00
19	.00	1560.00	.00	164.40	8.80	.00	.00
TOTAL:	126.00	2090.40	.00	2106.00	29.60	2.40	15.40

DATA SET 4

HOURS/DAY BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	.80	28.70	.00	18.20	.00	7.50	8.00
2	.00	.00	.00	.30	.00	.00	.30
3	.00	.00	.00	.00	4.10	.00	.30
4	.00	1.50	.00	1.60	.00	1.50	.00
5	.00	.00	.00	.00	.00	.00	.00
6	.00	2.80	.00	9.40	.00	.00	.00
7	.00	.00	.00	5.40	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.30
12	.00	.00	.00	.00	.00	.00	.00
13	.00	4.20	.00	4.60	.00	.00	.60
14	.00	23.00	.00	34.40	.00	.00	.00
15	.00	.00	.00	.50	.00	.00	.00
16	.00	3.20	.00	12.50	.00	.00	.00
17	.00	.00	.00	.30	.00	.00	.80
18	.00	.00	.00	.00	.00	.00	.30
19	.00	10.20	.00	.00	.70	1.30	.00
TOTAL:	.80	73.60	.00	87.20	4.80	10.30	10.60

TOTAL HOURS/YEAR BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	201.60	1492.40	.00	218.40	.00	15.00	8.00
2	.00	.00	.00	3.60	.00	.00	.30
3	.00	.00	.00	.00	16.40	.00	.30
4	.00	78.00	.00	19.20	.00	3.00	.00
5	.00	.00	.00	.00	.00	.00	.00
6	.00	145.60	.00	112.80	.00	.00	.00
7	.00	.00	.00	64.80	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.30
12	.00	.00	.00	.00	.00	.00	.00
13	.00	218.40	.00	55.20	.00	.00	.60
14	.00	1196.00	.00	412.80	.00	.00	.00
15	.00	.00	.00	6.00	.00	.00	.00
16	.00	166.40	.00	150.00	.00	.00	.00
17	.00	.00	.00	3.60	.00	.00	.80
18	.00	.00	.00	.00	.00	.00	.30
19	.00	530.40	.00	.00	2.80	2.60	.00
TOTAL:	201.60	3827.20	.00	1046.40	19.20	20.60	10.60

DEMAND DATA SET 5

HOURS/DAY BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	1.60	6.10	.00	50.70	12.10	12.60	4.30
2	.00	.00	.00	.00	.00	.00	.80
3	.00	.00	.00	.00	1.20	.00	1.40
4	.00	.00	.00	5.10	.30	.60	1.80
5	.00	.00	.00	.00	.00	.00	.00
6	.30	.00	.00	15.70	8.50	9.00	7.40
7	.00	.40	.00	5.10	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.80
12	.00	.00	.00	.00	.00	.00	.00
13	.00	.00	.00	.40	.60	1.30	.30
14	1.40	.00	.00	17.10	2.10	1.00	.60
15	.00	.00	.00	.00	.50	.00	.90
16	.00	.00	.00	5.00	.60	.00	1.90
17	.00	.00	.00	3.20	.00	1.50	6.50
18	.00	.00	.00	.00	.00	.00	.40
19	.00	2.90	.00	33.90	4.70	9.90	11.70
TOTAL:	3.30	9.40	.00	136.20	30.60	35.90	38.80

TOTAL HOURS/YEAR BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	403.20	317.20	.00	608.40	48.40	25.20	4.30
2	.00	.00	.00	.00	.00	.00	.80
3	.00	.00	.00	.00	4.80	.00	1.40
4	.00	.00	.00	61.20	1.20	1.20	1.80
5	.00	.00	.00	.00	.00	.00	.00
6	75.60	.00	.00	188.40	34.00	18.00	7.40
7	.00	20.80	.00	61.20	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.80
12	.00	.00	.00	.00	.00	.00	.00
13	.00	.00	.00	4.80	2.40	2.60	.30
14	352.80	.00	.00	205.20	8.40	2.00	.60
15	.00	.00	.00	.00	2.00	.00	.90
16	.00	.00	.00	60.00	2.40	.00	1.90
17	.00	.00	.00	38.40	.00	3.00	6.50
18	.00	.00	.00	.00	.00	.00	.40
19	.00	150.80	.00	406.80	18.80	19.80	11.70
TOTAL:	831.60	488.80	.00	1634.40	122.40	71.80	38.80

DEMAND DATA SET 6

HOURS/DAY BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	1.60	6.10	.00	50.70	12.10	12.60	4.30
2	.00	.00	.00	.00	.00	.00	.80
3	.00	.00	.00	.00	1.20	.00	1.40
4	.00	.00	.00	5.10	.30	.60	1.80
5	.00	.00	.00	.00	.00	.00	.00
6	.30	.00	.00	15.70	8.50	9.00	7.40
7	.00	.40	.00	5.10	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.80
12	.00	.00	.00	.00	.00	.00	.00
13	.00	.00	.00	.40	.60	1.30	.30
14	1.40	.00	.00	17.10	2.10	1.00	.60
15	.00	.00	.00	.00	.50	.00	.90
16	.00	.00	.00	5.00	.60	.00	1.90
17	.00	.00	.00	3.20	.00	1.50	6.50
18	.00	.00	.00	.00	.00	.00	.40
19	.00	1.40	.00	63.90	4.70	9.90	11.70
TOTAL:	3.30	7.90	.00	166.20	30.60	35.90	38.80

TOTAL HOURS/YEAR BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	403.20	317.20	.00	608.40	48.40	25.20	4.30
2	.00	.00	.00	.00	.00	.00	.80
3	.00	.00	.00	.00	4.80	.00	1.40
4	.00	.00	.00	61.20	1.20	1.20	1.80
5	.00	.00	.00	.00	.00	.00	.00
6	75.60	.00	.00	188.40	34.00	18.00	7.40
7	.00	20.80	.00	61.20	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.80
12	.00	.00	.00	.00	.00	.00	.00
13	.00	.00	.00	4.80	2.40	2.60	.30
14	352.80	.00	.00	205.20	8.40	2.00	.60
15	.00	.00	.00	.00	2.00	.00	.90
16	.00	.00	.00	60.00	2.40	.00	1.90
17	.00	.00	.00	38.40	.00	3.00	6.50
18	.00	.00	.00	.00	.00	.00	.40
19	.00	72.80	.00	766.80	18.80	19.80	11.70
TOTAL:	831.60	410.80	.00	1994.40	122.40	71.80	38.80

DEMAND DATA SET 7

HOURS/DAY BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	.70	6.30	.00	17.70	.00	3.60	13.40
2	.00	.00	.00	.00	.00	.00	.40
3	.00	.00	.00	.30	.00	.00	.00
4	.00	.00	3.90	.40	.00	.30	.50
5	.00	.00	.00	.00	.00	.00	.00
6	.00	1.50	.00	1.20	.00	1.80	8.20
7	.00	.00	.00	.40	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.40
12	.00	.00	.00	.00	.00	.00	.40
13	.00	.00	.00	.00	.00	.00	.80
14	.00	.70	.00	4.90	.00	.00	.00
15	.00	.00	.00	1.00	.00	.10	.30
16	.00	.00	.00	.70	.00	.30	.90
17	.00	.00	.00	.40	.00	.00	.00
18	.00	.00	.00	.00	.00	.00	.40
19	.00	8.40	.00	6.10	.00	1.90	8.30
TOTAL:	.70	16.90	3.90	33.10	.00	8.00	34.00

TOTAL HOURS/YEAR BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	176.40	327.60	.00	212.40	.00	7.20	13.40
2	.00	.00	.00	.00	.00	.00	.40
3	.00	.00	.00	3.60	.00	.00	.00
4	.00	.00	101.40	4.80	.00	.60	.50
5	.00	.00	.00	.00	.00	.00	.00
6	.00	78.00	.00	14.40	.00	3.60	8.20
7	.00	.00	.00	4.80	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.40
12	.00	.00	.00	.00	.00	.00	.40
13	.00	.00	.00	.00	.00	.00	.80
14	.00	36.40	.00	58.80	.00	.00	.00
15	.00	.00	.00	12.00	.00	.20	.30
16	.00	.00	.00	8.40	.00	.60	.90
17	.00	.00	.00	4.80	.00	.00	.00
18	.00	.00	.00	.00	.00	.00	.40
19	.00	436.80	.00	73.20	.00	3.80	8.30
TOTAL:	176.40	878.80	101.40	397.20	.00	16.00	34.00

DEMAND DATA SET 8

HOURS/DAY BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	.70	10.20	.00	17.70	.00	3.60	13.40
2	.00	.00	.00	.00	.00	.00	.40
3	.00	.00	.00	.30	.00	.00	.00
4	.00	.00	3.90	.40	.00	.30	.50
5	.00	.00	.00	.00	.00	.00	.00
6	.00	1.50	.00	1.20	.00	1.80	8.20
7	.00	.00	.00	.40	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.40
12	.00	.00	.00	.00	.00	.00	.40
13	.00	.00	.00	.00	.00	.00	.80
14	.00	.70	.00	4.90	.00	.00	.00
15	.00	.00	.00	1.00	.00	.10	.30
16	.00	.00	.00	.70	.00	.30	.90
17	.00	.00	.00	.40	.00	.00	.00
18	.00	.00	.00	.00	.00	.00	.40
19	.00	11.90	.00	16.10	.00	1.90	8.30
TOTAL:	.70	24.30	3.90	43.10	.00	8.00	34.00

TOTAL HOURS/YEAR BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	176.40	530.40	.00	212.40	.00	7.20	13.40
2	.00	.00	.00	.00	.00	.00	.40
3	.00	.00	.00	3.60	.00	.00	.00
4	.00	.00	101.40	4.80	.00	.60	.50
5	.00	.00	.00	.00	.00	.00	.00
6	.00	78.00	.00	14.40	.00	3.60	8.20
7	.00	.00	.00	4.80	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.40
12	.00	.00	.00	.00	.00	.00	.40
13	.00	.00	.00	.00	.00	.00	.80
14	.00	36.40	.00	58.80	.00	.00	.00
15	.00	.00	.00	12.00	.00	.20	.30
16	.00	.00	.00	8.40	.00	.60	.90
17	.00	.00	.00	4.80	.00	.00	.00
18	.00	.00	.00	.00	.00	.00	.40
19	.00	618.80	.00	193.20	.00	3.80	8.30
TOTAL:	176.40	1263.60	101.40	517.20	.00	16.00	34.00

DEMAND DATA SET 9

HOURS/DAY BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	1.70	12.20	.00	31.00	9.80	17.20	12.60
2	.00	.00	.00	.00	.00	.00	.00
3	.00	.00	.00	3.20	.00	.00	.00
4	.00	5.50	.00	37.70	.40	.40	4.40
5	.00	.00	.00	.00	.00	.00	.00
6	.00	12.40	.00	11.60	1.60	1.80	1.80
7	.00	1.60	.00	2.00	.00	.00	1.10
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.00
12	.00	.00	.00	.00	.00	.00	.00
13	.00	.00	.00	.00	.00	.00	.40
14	.00	1.10	.00	11.80	.40	1.50	2.20
15	.00	.00	.00	.00	.00	.00	1.00
16	.00	.00	.00	.40	.00	.00	6.50
17	.00	.00	.00	.00	.00	.00	10.80
18	.00	.00	.00	.00	.00	.00	.00
19	.00	.30	.00	1.80	.00	.00	12.00
TOTAL:	1.70	33.10	.00	99.50	12.20	20.90	52.80

TOTAL HOURS/YEAR BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	428.40	634.40	.00	372.00	39.20	34.40	12.60
2	.00	.00	.00	.00	.00	.00	.00
3	.00	.00	.00	38.40	.00	.00	.00
4	.00	286.00	.00	452.40	1.60	.80	4.40
5	.00	.00	.00	.00	.00	.00	.00
6	.00	644.80	.00	139.20	6.40	3.60	1.80
7	.00	83.20	.00	24.00	.00	.00	1.10
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.00
12	.00	.00	.00	.00	.00	.00	.00
13	.00	.00	.00	.00	.00	.00	.40
14	.00	57.20	.00	141.60	1.60	3.00	2.20
15	.00	.00	.00	.00	.00	.00	1.00
16	.00	.00	.00	4.80	.00	.00	6.50
17	.00	.00	.00	.00	.00	.00	10.80
18	.00	.00	.00	.00	.00	.00	.00
19	.00	15.60	.00	21.60	.00	.00	12.00
TOTAL:	428.40	1721.20	.00	1194.00	48.80	41.80	52.80

DEMAND DATA SET 10

HOURS/DAY BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	.70	.30	.00	6.10	.00	.20	4.80
2	.00	.00	.00	.00	.00	.00	.30
3	.00	.00	.00	.00	.00	.00	.00
4	.00	.00	.00	.90	.00	.00	.00
5	.00	.00	.00	.00	.00	.00	.00
6	.00	1.60	.00	4.20	.00	7.40	.00
7	.00	.00	.00	.40	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	1.80	.30
12	.00	.00	.00	.10	.00	.00	.00
13	.00	.00	.00	.00	.00	.00	.60
14	.20	.00	.00	9.00	.00	.50	.00
15	.00	.00	.00	.00	.00	.00	.00
16	.00	1.00	.00	2.90	.00	.00	.00
17	.00	.00	.00	7.30	.00	.00	.40
18	.00	.00	.00	.00	.00	.00	.30
19	.00	1.00	.00	23.60	.00	.00	.00
TOTAL:	.90	3.90	.00	54.50	.00	9.90	6.70

TOTAL HOURS/YEAR BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	176.40	15.60	.00	73.20	.00	.40	4.80
2	.00	.00	.00	.00	.00	.00	.30
3	.00	.00	.00	.00	.00	.00	.00
4	.00	.00	.00	10.80	.00	.00	.00
5	.00	.00	.00	.00	.00	.00	.00
6	.00	83.20	.00	50.40	.00	14.80	.00
7	.00	.00	.00	4.80	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	3.60	.30
12	.00	.00	.00	1.20	.00	.00	.00
13	.00	.00	.00	.00	.00	.00	.60
14	50.40	.00	.00	108.00	.00	1.00	.00
15	.00	.00	.00	.00	.00	.00	.00
16	.00	52.00	.00	34.80	.00	.00	.00
17	.00	.00	.00	87.60	.00	.00	.40
18	.00	.00	.00	.00	.00	.00	.30
19	.00	52.00	.00	283.20	.00	.00	.00
TOTAL:	226.80	202.80	.00	654.00	.00	19.80	6.70

DEMAND DATA SET 11

HOURS/DAY BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	1.70	2.70	.00	9.20	20.40	8.40	5.20
2	.00	.00	.00	.30	.00	.00	.40
3	.00	.00	.00	1.00	3.10	3.90	1.70
4	.00	.00	.00	.00	.30	.00	2.30
5	.00	.00	.00	.00	.00	.00	.00
6	.00	.90	.00	3.90	12.40	1.60	1.20
7	.00	.00	.00	1.40	.00	.00	.20
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.50	.70	.30	.30
12	.00	.00	.00	.00	.00	.00	.00
13	.00	.50	.00	3.10	.80	.00	1.00
14	.00	3.20	.00	8.00	.60	.10	.00
15	.00	.00	.00	.40	.00	.00	.00
16	.00	.00	.00	1.00	.00	.00	.00
17	.00	.00	.00	2.90	.00	.00	1.40
18	.00	.00	.00	.00	.00	.00	.00
19	.00	2.60	.00	2.60	1.20	1.20	.00
TOTAL:	1.70	9.90	.00	34.30	39.50	15.50	13.70

TOTAL HOURS/YEAR BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	428.40	140.40	.00	110.40	81.60	16.80	5.20
2	.00	.00	.00	3.60	.00	.00	.40
3	.00	.00	.00	12.00	12.40	7.80	1.70
4	.00	.00	.00	.00	1.20	.00	2.30
5	.00	.00	.00	.00	.00	.00	.00
6	.00	46.80	.00	46.80	49.60	3.20	1.20
7	.00	.00	.00	16.80	.00	.00	.20
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	6.00	2.80	.60	.30
12	.00	.00	.00	.00	.00	.00	.00
13	.00	26.00	.00	37.20	3.20	.00	1.00
14	.00	166.40	.00	96.00	2.40	.20	.00
15	.00	.00	.00	4.80	.00	.00	.00
16	.00	.00	.00	12.00	.00	.00	.00
17	.00	.00	.00	34.80	.00	.00	1.40
18	.00	.00	.00	.00	.00	.00	.00
19	.00	135.20	.00	31.20	4.80	2.40	.00
TOTAL:	428.40	514.80	.00	411.60	158.00	31.00	13.70

DEMAND DATA SET 12

HOURS/DAY BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	.00	.00	.00	.00	8.70	11.40	13.40
2	.00	.00	.00	.00	.00	.00	.00
3	.00	.00	.00	.00	.00	.00	.00
4	.00	.00	.00	.00	.00	.00	.00
5	.00	.00	.00	.00	.00	.00	.00
6	.00	.00	.00	.00	.00	.00	.00
7	.00	.00	.00	.00	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.00
12	.00	.00	.00	.00	.00	.00	.00
13	.00	.00	.00	.00	.00	.00	.00
14	.00	.00	.00	6.30	.00	.00	3.20
15	.00	.00	.00	.00	.00	.00	.00
16	.00	.00	.00	.00	.00	.00	.00
17	.00	.00	.00	.00	.00	.00	.00
18	.00	.00	.00	.00	.00	.00	.00
19	.00	.00	.00	.00	.00	.00	.00
TOTAL:	.00	.00	.00	6.30	8.70	11.40	16.60

TOTAL HOURS/YEAR BY FREQUENCY PERIOD

LOC	1	2	3	4	5	6	7
1	.00	.00	.00	.00	34.80	22.80	13.40
2	.00	.00	.00	.00	.00	.00	.00
3	.00	.00	.00	.00	.00	.00	.00
4	.00	.00	.00	.00	.00	.00	.00
5	.00	.00	.00	.00	.00	.00	.00
6	.00	.00	.00	.00	.00	.00	.00
7	.00	.00	.00	.00	.00	.00	.00
8	.00	.00	.00	.00	.00	.00	.00
9	.00	.00	.00	.00	.00	.00	.00
10	.00	.00	.00	.00	.00	.00	.00
11	.00	.00	.00	.00	.00	.00	.00
12	.00	.00	.00	.00	.00	.00	.00
13	.00	.00	.00	.00	.00	.00	.00
14	.00	.00	.00	75.60	.00	.00	3.20
15	.00	.00	.00	.00	.00	.00	.00
16	.00	.00	.00	.00	.00	.00	.00
17	.00	.00	.00	.00	.00	.00	.00
18	.00	.00	.00	.00	.00	.00	.00
19	.00	.00	.00	.00	.00	.00	.00
TOTAL:	.00	.00	.00	75.60	34.80	22.80	16.60

APPENDIX D: SAMPLE COMPUTER OUTPUT

SUMMARY OF INPUT DATA

```
TOTAL ANNUAL WORKING DAYS PER INSPECTOR:  218
TOTAL ANNUAL WORK DAYS REQUIRED:  252
INSPECTOR HOURS AVAILABLE PER DAY:    6.50

TOTAL NUMBER OF NODES IN NETWORK:  19
     DEPOT  1 AT NODE  1
     DEPOT  2 AT NODE 14

NUMBER OF NON-DAILY FREQUENCY PERIODS:   6
     PERIOD  1 -- FREQUENCY:  52 PER YEAR
     PERIOD  2 -- FREQUENCY:  26 PER YEAR
     PERIOD  3 -- FREQUENCY:  12 PER YEAR
     PERIOD  4 -- FREQUENCY:   4 PER YEAR
     PERIOD  5 -- FREQUENCY:   2 PER YEAR
     PERIOD  6 -- FREQUENCY:   1 PER YEAR
```

SUMMARY OF DEMAND LOADING

```
ANNUAL FREQUENCY OF 252:
          TOTAL DEMAND =  3200.40 HOURS PER YEAR
ANNUAL FREQUENCY OF  52:
          TOTAL DEMAND =   338.00 HOURS PER YEAR
ANNUAL FREQUENCY OF  26:
          TOTAL DEMAND =      .00 HOURS PER YEAR
ANNUAL FREQUENCY OF  12:
          TOTAL DEMAND =   747.60 HOURS PER YEAR
ANNUAL FREQUENCY OF   4:
          TOTAL DEMAND =    41.60 HOURS PER YEAR
ANNUAL FREQUENCY OF   2:
          TOTAL DEMAND =    16.20 HOURS PER YEAR
ANNUAL FREQUENCY OF   1:
          TOTAL DEMAND =    28.20 HOURS PER YEAR

TOTAL ANNUAL DAILY DEMAND =        3200.40 HOURS
TOTAL ANNUAL NON-DAILY DEMAND =    1171.60 HOURS
GRAND TOTAL OF ANNUAL DEMAND =     4372.00 HOURS
```

SUMMARY OF INSPECTOR UTILIZATION

LOWER BOUND ON MINIMUM NUMBER OF INSPECTORS: 4
ANNUAL INSPECTOR AVAILABILITY IN HOURS: 1417.00

**** TOTAL OF 4 INSPECTORS ASSIGNED ****

```
INSPECTOR  1 BASED AT NODE  1:
     TOTAL TIME INSPECTING = 1417.00    UTILIZATION = 100.00%
     TOTAL TIME TRAVELLING =     .00    UTILIZATION =    .00%
     SLACK HOURS REMAINING =     .00    UTILIZATION = 100.00%

INSPECTOR  2 BASED AT NODE  1:
     TOTAL TIME INSPECTING = 1176.90    UTILIZATION =  83.06%
     TOTAL TIME TRAVELLING =  181.92    UTILIZATION =  12.84%
     SLACK HOURS REMAINING =   58.18    UTILIZATION =  95.89%

INSPECTOR  3 BASED AT NODE  1:
     TOTAL TIME INSPECTING =  503.90    UTILIZATION =  35.56%
     TOTAL TIME TRAVELLING =  130.25    UTILIZATION =   9.19%
     SLACK HOURS REMAINING =  782.85    UTILIZATION =  44.75%

INSPECTOR  4 BASED AT NODE 14:
     TOTAL TIME INSPECTING = 1274.20    UTILIZATION =  89.92%
     TOTAL TIME TRAVELLING =  130.83    UTILIZATION =   9.23%
     SLACK HOURS REMAINING =   11.97    UTILIZATION =  99.16%

TOTALS FOR ALL INSPECTORS:
     AVERAGE TIME SPENT INSPECTING =  77.13%
     AVERAGE TIME SPENT TRAVELLING =   7.82%
     AVERAGE SLACK TIME UTILIZED =    84.95%

NUMBER OF INSPECTORS ASSIGNED TO EACH DEPOT
     DEPOT AT NODE  1:   3 INSPECTORS
     DEPOT AT NODE 14:   1 INSPECTORS
```

SUMMARY OF DAILY ASSIGNMENTS FOR DEPOT 1

LOC	INSP	FIRST DAY	LAST DAY	TIME	TOTAL DAYS	DEM HRS
1	1	1	218	6.50	218	1417.00
1	2	219	252	6.50	34	221.00
1	2	1	184	1.80	184	331.20
1	3	185	252	1.80	68	122.40
6	2	1	184	1.60	184	294.40
6	3	185	252	1.60	68	108.80

SLACK SUMMARY AFTER DAILY ASSIGNMENTS FOR DEPOT 1

(II)	INSP	BEG	END	SLACK	DAYS
1	1	1	252	.00	252
2	2	1	184	2.77	184
3	2	185	252	.00	68
4	3	1	150	6.50	150
5	3	151	184	.00	34
6	3	185	252	2.77	68

******* INSPECTORS WITH SLACK OVER TOTAL WORKING PERIOD *******

I	INSP	DAYS AVAIL	HRS AVAIL	INDEX
1	2	184	2.77	2
2	3	68	2.77	6

EQUIVALENT AVAILABILITIES IN DAYS:

FREQUENCY CLASS

INSP	1	2	3	4	5	6	7	8	9	10
2	184	38	1°	9	3	2	1			
3	68	14	7	3	1	0	0			

NON-DAILY ASSIGNMENTS FOR NODES VISITED DAILY FOR DEPOT 1

LOC	INSP	FIRST DAY LAST DAY	TIME	TOTAL DAYS	DEM HRS
6	2	.....FROM SLACK.....	.50	38	19.00
6	3	.....FROM SLACK.....	.50	14	7.00
1	2	.....FROM SLACK.....	8.70	9	78.30
1	3	.....FROM SLACK.....	8.70	3	26.10
6	2	.....FROM SLACK.....	4.60	9	41.40
6	3	.....FROM SLACK.....	4.60	3	13.80
1	2	.....FROM SLACK.....	6.00	3	18.00
1	3	.....FROM SLACK.....	6.00	1	6.00
6	2	.....FROM SLACK.....	3.20	3	9.60
6	3	.....FROM SLACK.....	3.20	1	3.20
1	2	.....FROM SLACK.....	.90	2	1.80
6	2	.....FROM SLACK.....	3.90	2	7.80
1	2	.....FROM SLACK.....	4.80	1	4.80
6	2	.....FROM SLACK.....	1.40	1	1.40

SLACK AFTER SATISFYING CYCLIC DEMANDS OF NODES VISITED DAILY FOR DEPOT 1

(II)	INSP	BEG	END	SLACK	DAYS
1	1	1	252	.00	252
2	2	1	184	1.78	184
3	2	185	252	.00	68
4	3	1	150	6.50	150
5	3	151	184	.00	34
6	3	185	252	1.94	68

ASSIGNMENTS MADE TO INSPECTORS WITH SUFFICIENT SLACK TIME

LOC	INSP	FIRST DAY LAST DAY	TIME	TOTAL DAYS	DEM HRS
4	2	.....FROM SLACK.....	.80	38	30.40
4	3	.....FROM SLACK.....	.80	14	11.20
11	2	.....FROM SLACK.....	.60	38	22.80
11	3	.....FROM SLACK.....	.60	14	8.40
4	2	.....FROM SLACK.....	2.60	9	23.40
4	3	.....FROM SLACK.....	2.60	3	7.80
11	2	.....FROM SLACK.....	.80	9	7.20
11	3	.....FROM SLACK.....	.80	3	2.40
4	2	.....FROM SLACK.....	2.10	2	4.20
4	2	.....FROM SLACK.....	.40	1	.40
11	2	.....FROM SLACK.....	2.40	1	2.40
2	2	.....FROM SLACK.....	1.20	9	10.80
2	3	.....FROM SLACK.....	1.20	3	3.60
3	2	.....FROM SLACK.....	.60	9	5.40
3	3	.....FROM SLACK.....	.60	3	1.80
7	2	.....FROM SLACK.....	2.60	9	23.40
7	3	.....FROM S' ACK.....	2.60	3	7.80

13	2	.....FROM SLACK.....	1.50	9	13.50
13	3	.....FROM SLACK.....	1.50	3	4.50
2	2	.....FROM SLACK.....	.70	1	.70
3	2	.....FROM SLACK.....	2.30	1	2.30
7	2	.....FROM SLACK.....	.30	1	.30
13	2	.....FROM SLACK.....	.60	1	.60
12	2	.....FROM SLACK.....	.40	1	.40

SLACK AFTER SATISFYING ALL CYCLIC DEMANDS FOR NODES OF DEPOT 1

(II)	INSP	BEG	END	SLACK	DAYS
1	1	1	252	.00	252
2	2	1	184	.32	184
3	2	185	252	.00	68
4	3	1	150	6.50	150
5	3	151	184	.00	34
6	3	185	252	.61	68

SUMMARY OF DAILY ASSIGNMENTS FOR DEPOT 14

LOC	INSP	FIRST DAY	LAST DAY	TIME	TOTAL DAYS	DEM HRS
14	4	1	116	2.00	116	232.00
14	3	117	150	2.00	34	68.00
14	4	151	252	2.00	102	204.00
19	4	1	116	.80	116	92.80
19	3	117	150	.80	34	27.20
19	4	151	252	.80	102	81.60

SLACK SUMMARY AFTER DAILY ASSIGNMENTS FOR DEPOT 14

(II)	INSP	BEG	END	SLACK	DAYS
1	1	1	252	.00	252
2	2	1	184	.32	184
3	2	185	252	.00	68
4	3	1	116	6.50	116
5	3	117	150	2.37	34
6	3	151	184	.00	34
7	3	185	252	.61	68
8	4	1	116	3.37	116
9	4	117	150	.00	34
10	4	151	252	3.37	102

******* INSPECTORS WITH SLACK OVER TOTAL WORKING PERIOD *******

I	INSP	DAYS AVAIL	HRS AVAIL	INDEX
1	4	116	3.37	8
2	4	102	3.37	10
3	3	34	2.37	5

EQUIVALENT AVAILABILITIES IN DAYS:

FREQUENCY CLASS

INSP	1	2	3	4	5	6	7	8	9	10
4	116	24	12	6	2	1	1			
4	102	22	11	5	2	1	0			
3	34	6	3	1	0	0	0			

NON-DAILY ASSIGNMENTS FOR NODES VISITED DAILY FOR DEPOT 14

LOC	INSP	FIRST DAY	LAST DAY	TIME	TOTAL DAYS	DEM HRS
14	4	.....FROM	SLACK.....	.70	24	16.80
14	4	.....FROM	SLACK.....	.70	22	15.40
14	3	.....FROM	SLACK.....	.70	6	4.20
19	4	.....FROM	SLACK.....	2.50	24	60.00
19	4	.....FROM	SLACK.....	2.50	22	55.00
19	3	.....FROM	SLACK.....	2.50	6	15.00
14	4	.....FROM	SLACK.....	7.70	6	46.20
14	4	.....FROM	SLACK.....	7.70	5	38.50
14	3	.....FROM	SLACK.....	7.70	1	7.70
19	4	.....FROM	SLACK.....	15.20	6	91.20
19	4	.....FROM	SLACK.....	15.20	5	76.00
19	3	.....FROM	SLACK.....	15.20	1	15.20
19	4	.....FROM	SLACK.....	1.20	2	2.40
19	4	.....FROM	SLACK.....	1.20	2	2.40
14	4	.....FROM	SLACK.....	1.20	1	1.20
14	4	.....FROM	SLACK.....	1.20	1	1.20
14	4	.....FROM	SLACK.....	4.90	1	4.90
19	4	.....FROM	SLACK.....	8.40	1	8.40

SLAC AFTER SATISFYING CYCLIC DEMANDS OF NODES VISITED DAILY FOR DEPOT 14

(II)	INSP	BEG	END	SLACK	DAYS
1	1	1	252	.00	252
2	2	1	184	.32	184
3	2	185	252	.00	68
4	3	1	116	6.50	116
5	3	117	150	1.13	34
6	3	151	184	.00	34
7	3	185	252	.61	68
8	4	1	116	1.37	116
9	4	117	150	.00	34
10	4	151	252	1.52	102

** INSPECTOR 4 INSUFFICIENT SLACK **
SHORTFALL ATTEMPTING TO SATISFY NODE 18 PERIOD 4 DEMAND

******* INSPECTORS WITH SLACK OVER TOTAL WORKING PERIOD *******

I	INSP	DAYS AVAIL	HRS AVAIL	INDEX
1	3	116	6.50	4
2	4	102	.17	10
3	3	34	.08	5

EQUIVALENT AVAILABILITIES IN DAYS:

FREQUENCY CLASS

INSP	1	2	3	4	5	6	7	8	9	10
3	116	24	12	6	2	1	1			
4	102	22	11	5	2	1	0			
3	34	6	3	1	0	0	0			

ASSIGNMENTS MADE TO INSPECTORS WITH SUFFICIENT SLACK TIME

LOC	INSP	FIRST DAY	LAST DAY	TIME	TOTAL DAYS	DEM HRS
16	4	.....FROM	SLACK.....	.60	24	14.40
16	4	.....FROM	SLACK.....	.60	22	13.20
16	3	.....FROM	SLACK.....	.60	6	3.60
17	4	.....FROM	SLACK.....	.80	24	19.20
17	4	.....FROM	SLACK.....	.80	22	17.60
17	3	.....FROM	SLACK.....	.80	6	4.80
16	4	.....FROM	SLACK.....	1.40	6	8.40
16	4	.....FROM	SLACK.....	1.40	5	7.00

16	3	.....FROM SLACK.....	1.40	1	1.40
17	4	.....FROM SLACK.....	13.80	6	82.80
17	4	.....FROM SLACK.....	13.80	5	69.00
17	3	.....FROM SLACK.....	13.80	1	13.80
17	4	.....FROM SLACK.....	1.00	1	1.00
15	4	.....FROM SLACK.....	.60	6	3.60
15	4	.....FROM SLACK.....	.60	5	3.00
15	3	.....FROM SLACK.....	.60	1	.60
18	3	.....FROM SLACK.....	1.00	6	6.00
18	4	.....FROM SLACK.....	1.00	5	5.00
18	3	.....FROM SLACK.....	1.00	1	1.00
18	3	.....FROM SLACK.....	.60	1	.60

SLACK AFTER SATISFYING ALL CYCLIC DEMANDS FOR NODES OF DEPOT 14

(II)	INSP	BEG	END	SLACK	DAYS
1	1	1	252	.00	252
2	2	1	184	.32	184
3	2	185	252	.00	68
4	3	1	116	6.37	116
5	3	117	150	.05	34
6	3	151	184	.00	34
7	3	185	252	.61	68
8	4	1	116	.00	116
9	4	117	150	.00	34
10	4	151	252	.12	102

************ FINAL SLACK SUMMARY ************

(II)	INSP	BEG	END	SLACK	DAYS
1	1	1	252	.00	252
2	2	1	184	.32	184
3	2	185	252	.00	68
4	3	1	116	6.37	116
5	3	117	150	.05	34
6	3	151	184	.00	34
7	3	185	252	.61	68
8	4	1	116	.00	116
9	4	117	150	.00	34
10	4	151	252	.12	102

```
************ SUMMARY OF TOURS ************

***** TOURS OF INSPECTOR  1 *****


***** TOURS OF INSPECTOR  2 *****

   TOUR NUMBER  1

           NODE  1      TIME SPENT     .00 HOURS
           NODE  6      TIME SPENT    1.60 HOURS
           NODE  1      TIME SPENT     .00 HOURS

        TOUR LENGTH:  1.93    MAXIMUM TIME OF TOUR:  4.70
        SLACK DAYS WHEN TOUR POSSIBLE:     1 TO 184    DAYS REQUIRED: 119

        TOUR UTILIZATION FROM SLACK TIME:

           NODE   TIME SPENT   DAYS   FREQ PERIOD

             6         .50      38         2
             6        4.60       9         4
             6        3.20       3         5
             6        3.90       2         6
             6        1.40       1         7

   TOUR NUMBER  2

           NODE  1      TIME SPENT     .00 HOURS
           NODE  4      TIME SPENT     .80 HOURS
           NODE  6      TIME SPENT    1.60 HOURS
           NODE  1      TIME SPENT     .00 HOURS

        TOUR LENGTH:  3.23    MAXIMUM TIME OF TOUR:  4.70
        SLACK DAYS WHEN TOUR POSSIBLE:     1 TO 184    DAYS REQUIRED:  38

        TOUR UTILIZATION FROM SLACK TIME:

           NODE   TIME SPENT   DAYS   FREQ PERIOD

             4        2.60       9         4
             4        2.10       2         6
             4         .40       1         7

   TOUR NUMBER  3

           NODE  1      TIME SPENT     .00 HOURS
           NODE 11      TIME SPENT     .60 HOURS
           NODE  1      TIME SPENT     .00 HOURS

        TOUR LENGTH:  2.10    MAXIMUM TIME OF TOUR:  6.50
        SLACK DAYS WHEN TOUR POSSIBLE:     1 TO 184    DAYS REQUIRED:  37
```

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
11	.80	9	4
11	2.40	1	7

TOUR NUMBER 4

NODE 1	TIME SPENT	.00	HOURS
NODE 2	TIME SPENT	1.20	HOURS
NODE 4	TIME SPENT	.00	HOURS
NODE 6	TIME SPENT	1.60	HOURS
NODE 1	TIME SPENT	.00	HOURS

TOUR LENGTH: 4.05 MAXIMUM TIME OF TOUR: 4.70
SLACK DAYS WHEN TOUR POSSIBLE: 1 TO 184 DAYS REQUIRED: 9

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
2	.70	1	7

TOUR NUMBER 5

NODE 1	TIME SPENT	.00	HOURS
NODE 3	TIME SPENT	.60	HOURS
NODE 4	TIME SPENT	.00	HOURS
NODE 6	TIME SPENT	1.60	HOURS
NODE 1	TIME SPENT	.00	HOURS

TOUR LENGTH: 4.37 MAXIMUM TIME OF TOUR: 4.70
SLACK DAYS WHEN TOUR POSSIBLE: 1 TO 184 DAYS REQUIRED: 9

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
3	2.30	1	7

TOUR NUMBER 6

NODE 1	TIME SPENT	.00	HOURS
NODE 7	TIME SPENT	2.60	HOURS
NODE 6	TIME SPENT	1.60	HOURS
NODE 1	TIME SPENT	.00	HOURS

TOUR LENGTH: 4.70 MAXIMUM TIME OF TOUR: 4.70
SLACK DAYS WHEN TOUR POSSIBLE: 1 TO 184 DAYS REQUIRED: 9

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
7	.30	1	7

TOUR NUMBER 7

NODE 1 TIME SPENT .00 HOURS
NODE 13 TIME SPENT 1.50 HOURS
NODE 1 TIME SPENT .00 HOURS

TOUR LENGTH: 3.50 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 1 TO 184 DAYS REQUIRED: 9

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
13	.60	1	7

TOUR NUMBER 8

NODE 1 TIME SPENT .00 HOURS
NODE 12 TIME SPENT .40 HOURS
NODE 11 TIME SPENT .60 HOURS
NODE 1 TIME SPENT .00 HOURS

TOUR LENGTH: 2.83 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 1 TO 184 DAYS REQUIRED: 1

TOUR UTILIZATION FROM SLACK TIME:

NODE TIME SPENT DAYS FREQ PERIOD

--NONE--

***** TOURS OF INSPECTOR 3 *****

TOUR NUMBER 1

NODE 1 TIME SPENT .00 HOURS
NODE 6 TIME SPENT 1.60 HOURS
NODE 1 TIME SPENT .00 HOURS

TOUR LENGTH: 1.93 MAXIMUM TIME OF TOUR: 4.70
SLACK DAYS WHEN TOUR POSSIBLE: 185 TO 252 DAYS REQUIRED: 45

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
6	.50	14	2
6	4.60	3	4
6	3.20	1	5

TOUR NUMBER 2

NODE 1 TIME SPENT .00 HOURS
NODE 4 TIME SPENT .80 HOURS
NODE 6 TIME SPENT 1.60 HOURS
NODE 1 TIME SPENT .00 HOURS

TOUR LENGTH: 3.23 MAXIMUM TIME OF TOUR: 4.70
SLACK DAYS WHEN TOUR POSSIBLE: 185 TO 252 DAYS REQUIRED: 14

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
4	2.60	3	4

TOUR NUMBER 3

NODE 1 TIME SPENT .00 HOURS
NODE 11 TIME SPENT .60 HOURS
NODE 1 TIME SPENT .00 HOURS

TOUR LENGTH: 2.10 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 185 TO 252 DAYS REQUIRED: 14

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
11	.80	3	4

TOUR NUMBER 4

NODE 1 TIME SPENT .00 HOURS
NODE 2 TIME SPENT 1.20 HOURS
NODE 4 TIME SPENT .00 HOURS
NODE 6 TIME SPENT 1.60 HOURS
NODE 1 TIME SPENT .00 HOURS

TOUR LENGTH: 4.05 MAXIMUM TIME OF TOUR: 4.70
SLACK DAYS WHEN TOUR POSSIBLE: 185 TO 252 DAYS REQUIRED: 3

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD

--NONE--

TOUR NUMBER 5

NODE 1	TIME SPENT	.00 HOURS
NODE 3	TIME SPENT	.60 HOURS
NODE 4	TIME SPENT	.00 HOURS
NODE 6	TIME SPENT	1.60 HOURS
NODE 1	TIME SPENT	.00 HOURS

TOUR LENGTH: 4.37 MAXIMUM TIME OF TOUR: 4.70
SLACK DAYS WHEN TOUR POSSIBLE: 185 TO 252 DAYS REQUIRED: 3

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD

--NONE--

TOUR NUMBER 6

NODE 1	TIME SPENT	.00 HOURS
NODE 7	TIME SPENT	2.60 HOURS
NODE 6	TIME SPENT	1.60 HOURS
NODE 1	TIME SPENT	.00 HOURS

TOUR LENGTH: 4.70 MAXIMUM TIME OF TOUR: 4.70
SLACK DAYS WHEN TOUR POSSIBLE: 185 TO 252 DAYS REQUIRED: 3

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD

--NONE--

TOUR NUMBER 7

NODE 1	TIME SPENT	.00 HOURS
NODE 13	TIME SPENT	1.50 HOURS
NODE 1	TIME SPENT	.00 HOURS

TOUR LENGTH: 3.50 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 185 TO 252 DAYS REQUIRED: 3

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD

--NONE--

TOUR NUMBER 8

NODE 1	TIME SPENT	.00 HOURS
NODE 14	TIME SPENT	.00 HOURS
NODE 19	TIME SPENT	.80 HOURS
NODE 14	TIME SPENT	2.00 HOURS
NODE 1	TIME SPENT	.00 HOURS

TOUR LENGTH: 4.13 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 117 TO 150 DAYS REQUIRED: 20

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
14	.70	6	2
19	2.50	6	2
14	7.70	1	4
19	15.20	1	4

TOUR NUMBER 9

NODE 1	TIME SPENT	.00 HOURS
NODE 14	TIME SPENT	.00 HOURS
NODE 16	TIME SPENT	.60 HOURS
NODE 14	TIME SPENT	.00 HOURS
NODE 19	TIME SPENT	.80 HOURS
NODE 14	TIME SPENT	2.00 HOURS
NODE 1	TIME SPENT	.00 HOURS

TOUR LENGTH: 5.73 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 117 TO 150 DAYS REQUIRED: 6

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
16	1.40	1	4

TOUR NUMBER 10

NODE 1	TIME SPENT	.00 HOURS
NODE 14	TIME SPENT	.00 HOURS
NODE 17	TIME SPENT	.80 HOURS
NODE 14	TIME SPENT	.00 HOURS
NODE 19	TIME SPENT	.80 HOURS
NODE 14	TIME SPENT	2.00 HOURS
NODE 1	TIME SPENT	.00 HOURS

TOUR LENGTH: 5.60 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 117 TO 150 DAYS REQUIRED: 6

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
17	13.80	1	4

TOUR NUMBER 11

NODE 1	TIME SPENT	.00	HOURS
NODE 14	TIME SPENT	.00	HOURS
NODE 16	TIME SPENT	.00	HOURS
NODE 15	TIME SPENT	.60	HOURS
NODE 16	TIME SPENT	.00	HOURS
NODE 14	TIME SPENT	.00	HOURS
NODE 19	TIME SPENT	.80	HOURS
NODE 14	TIME SPENT	2.00	HOURS
NODE 1	TIME SPENT	.00	HOURS

TOUR LENGTH: 6.07 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 117 TO 150 DAYS REQUIRED: 1

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
	--NONE--		

TOUR NUMBER 12

NODE 1	TIME SPENT	.00	HOURS
NODE 14	TIME SPENT	.00	HOURS
NODE 18	TIME SPENT	1.00	HOURS
NODE 14	TIME SPENT	.00	HOURS
NODE 1	TIME SPENT	.00	HOURS

TOUR LENGTH: 2.33 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 1 TO 116 DAYS REQUIRED: 6

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
18	.60	1	7

TOUR NUMBER 13

```
        NODE  1      TIME SPENT     .00 HOURS
        NODE 14      TIME SPENT     .00 HOURS
        NODE 18      TIME SPENT    1.00 HOURS
        NODE 19      TIME SPENT     .80 HOURS
        NODE 14      TIME SPENT    2.00 HOURS
        NODE  1      TIME SPENT     .00 HOURS
```

TOUR LENGTH: 5.30 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 117 TO 150 DAYS REQUIRED: 1

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD

--NONE--

***** TOURS OF INSPECTOR 4 *****

TOUR NUMBER 1

```
        NODE 14      TIME SPENT     .00 HOURS
        NODE 19      TIME SPENT     .80 HOURS
        NODE 14      TIME SPENT     .00 HOURS
```

TOUR LENGTH: 1.13 MAXIMUM TIME OF TOUR: 4.50
SLACK DAYS WHEN TOUR POSSIBLE: 1 TO 116 DAYS REQUIRED: 116

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
19	2.50	24	2
19	15.20	6	4
19	1.20	2	5
19	8.40	1	7

TOUR NUMBER 2

```
        NODE 14      TIME SPENT     .00 HOURS
        NODE 19      TIME SPENT     .80 HOURS
        NODE 14      TIME SPENT     .00 HOURS
```

TOUR LENGTH: 1.13 MAXIMUM TIME OF TOUR: 4.50
SLACK DAYS WHEN TOUR POSSIBLE: 151 TO 252 DAYS REQUIRED: 97

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
19	2.50	22	2
19	15.20	5	4
19	1.20	2	5

TOUR NUMBER 3

NODE 14 TIME SPENT .00 HOURS
NODE 17 TIME SPENT .80 HOURS
NODE 16 TIME SPENT .60 HOURS
NODE 14 TIME SPENT .00 HOURS

TOUR LENGTH: 2.57 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 1 TO 116 DAYS REQUIRED: 18

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
16	1.40	6	4
17	13.80	6	4
17	1.00	1	7

TOUR NUMBER 4

NODE 14 TIME SPENT .00 HOURS
NODE 17 TIME SPENT .80 HOURS
NODE 16 TIME SPENT .60 HOURS
NODE 14 TIME SPENT .00 HOURS

TOUR LENGTH: 2.57 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 151 TO 252 DAYS REQUIRED: 17

TOUR UTILIZATION FROM SLACK TIME:

NODE	TIME SPENT	DAYS	FREQ PERIOD
16	1.40	5	4
17	13.80	5	4

TOUR NUMBER 5

NODE 14 TIME SPENT .00 HOURS
NODE 17 TIME SPENT .80 HOURS
NODE 16 TIME SPENT .00 HOURS
NODE 15 TIME SPENT .60 HOURS
NODE 16 TIME SPENT .60 HOURS
NODE 14 TIME SPENT .00 HOURS

TOUR LENGTH: 3.50 MAXIMUM TIME OF TOUR: 6.50
SLACK DAYS WHEN TOUR POSSIBLE: 1 TO 116 DAYS REQUIRED: 6

```
    TOUR UTILIZATION FROM SLACK TIME:

        NODE    TIME SPENT    DAYS    FREQ PERIOD

                --NONE--

TOUR NUMBER  6

        NODE 14      TIME SPENT     .00 HOURS
        NODE 17      TIME SPENT     .80 HOURS
        NODE 16      TIME SPENT     .00 HOURS
        NODE 15      TIME SPENT     .60 HOURS
        NODE 16      TIME SPENT     .60 HOURS
        NODE 14      TIME SPENT     .00 HOURS

    TOUR LENGTH:   3.50     MAXIMUM TIME OF TOUR:  6.50
    SLACK DAYS WHEN TOUR POSSIBLE:  151 TO 252     DAYS REQUIRED:    5

    TOUR UTILIZATION FROM SLACK TIME:

        NODE    TIME SPENT    DAYS    FREQ PERIOD

                --NONE--

TOUR NUMBER  7

        NODE 14      TIME SPENT     .00 HOURS
        NODE 18      TIME SPENT    1.00 HOURS
        NODE 19      TIME SPENT     .80 HOURS
        NODE 14      TIME SPENT     .00 HOURS

    TOUR LENGTH:   2.30     MAXIMUM TIME OF TOUR:  4.50
    SLACK DAYS WHEN TOUR POSSIBLE:  151 TO 252     DAYS REQUIRED:    5

    TOUR UTILIZATION FROM SLACK TIME:

        NODE    TIME SPENT    DAYS    FREQ PERIOD

                --NONE--
```

**** SUMMARY OF DEMAND SATISFACTION BY INSPECTOR ****

NODE 1:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
1	1	DEP	6.50	218	1417.00
1	2	DEP	6.50	34	221.00
1	2	DEP	1.80	184	331.20
1	3	DEP	1.80	68	122.40
				TOTAL:	2091.60
4	2	DEP	8.70	9	78.30
4	3	DEP	8.70	3	26.10
				TOTAL:	104.40
5	2	DEP	6.00	3	18.00
5	3	DEP	6.00	1	6.00
				TOTAL:	24.00
6	2	DEP	.90	2	1.80
				TOTAL:	1.80
7	2	DEP	4.80	1	4.80
				TOTAL:	4.80

NODE 2:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
4	2	4	1.20	9	10.80
4	3	4	1.20	3	3.60
				TOTAL:	14.40
7	2	4	.70	1	.70
				TOTAL:	.70

NODE 3:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
4	2	5	.60	9	5.40
4	3	5	.60	3	1.80
				TOTAL:	7.20
7	2	5	2.30	1	2.30
				TOTAL:	2.30

NODE 4:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
2	2	2	.80	38	30.40
2	3	2	.80	14	11.20
				TOTAL:	41.60
4	2	2	2.60	9	23.40
4	3	2	2.60	3	7.80
				TOTAL:	31.20
6	2	2	2.10	2	4.20
				TOTAL:	4.20
7	2	2	.40	1	.40
				TOTAL:	.40

NODE 5:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS

NODE 6:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
1	2	1	1.60	184	294.40
1	3	1	1.60	68	108.80
				TOTAL:	403.20
2	2	1	.50	38	19.00
2	3	1	.50	14	7.00
				TOTAL:	26.00
4	2	1	4.60	9	41.40
4	3	1	4.60	3	13.80
				TOTAL:	55.20
5	2	1	3.20	3	9.60
5	3	1	3.20	1	3.20
				TOTAL:	12.80
6	2	1	3.90	2	7.80
				TOTAL:	7.80
7	2	1	1.40	1	1.40
				TOTAL:	1.40

NODE 7:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
4	2	6	2.60	9	23.40
4	3	6	2.60	3	7.80
				TOTAL:	31.20
7	2	6	.30	1	.30
				TOTAL:	.30

NODE 8:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS

NODE 9:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS

NODE 10:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS

NODE 11:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
2	2	3	.60	38	22.80
2	3	3	.60	14	8.40
				TOTAL:	31.20
4	2	3	.80	9	7.20
4	3	3	.80	3	2.40
				TOTAL:	9.60
7	2	3	2.40	1	2.40
				TOTAL:	2.40

NODE 12:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
7	2	8	.40	1	.40
				TOTAL:	.40

NODE 13:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
4	2	7	1.50	9	13.50
4	3	7	1.50	3	4.50
				TOTAL:	18.00
7	2	7	.60	1	.60
				TOTAL:	.60

NODE 14:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
1	4	DEP	2.00	116	232.00
1	3	8	2.00	34	68.00
1	4	DEP	2.00	102	204.00
				TOTAL:	504.00
2	4	DEP	.70	24	16.80
2	4	DEP	.70	22	15.40
2	3	8	.70	6	4.20
				TOTAL:	36.40
4	4	DEP	7.70	6	46.20
4	4	DEP	7.70	5	38.50
4	3	8	7.70	1	7.70
				TOTAL:	92.40
6	4	DEP	1.20	1	1.20
6	4	DEP	1.20	1	1.20
				TOTAL:	2.40
7	4	DEP	4.90	1	4.90
				TOTAL:	4.90

NODE 15:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
4	4	5	.60	6	3.60
4	4	6	.60	5	3.00
4	3	11	.60	1	.60
				TOTAL:	7.20

NODE 16:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
2	4	3	.60	24	14.40
2	4	4	.60	22	13.20
2	3	9	.60	6	3.60
				TOTAL:	31.20
4	4	3	1.40	6	8.40
4	4	4	1.40	5	7.00
4	3	9	1.40	1	1.40
				TOTAL:	16.80

NODE 17:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
2	4	3	.80	24	19.20
2	4	4	.80	22	17.60
2	3	10	.80	6	4.80
				TOTAL:	41.60
4	4	3	13.80	6	82.80
4	4	4	13.80	5	69.00
4	3	10	13.80	1	13.80
				TOTAL:	165.60
7	4	3	1.00	1	1.00
				TOTAL:	1.00

NODE 18:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
4	3	12	1.00	6	6.00
4	4	7	1.00	5	5.00
4	3	13	1.00	1	1.00
				TOTAL:	12.00
7	3	12	.60	1	.60
				TOTAL:	.60

NODE 19:

FRQ	INSP	REF TOUR	TIME PER VIS	DAYS	DEM HRS
1	4	1	.80	116	92.80
1	3	8	.80	34	27.20
1	4	2	.80	102	81.60
				TOTAL:	201.60
2	4	1	2.50	24	60.00
2	4	2	2.50	22	55.00
2	3	8	2.50	6	15.00
				TOTAL:	130.00
4	4	1	15.20	6	91.20
4	4	2	15.20	5	76.00
4	3	8	15.20	1	15.20
				TOTAL:	182.40
5	4	1	1.20	2	2.40
5	4	2	1.20	2	2.40
				TOTAL:	4.80
7	4	1	8.40	1	8.40
				TOTAL:	8.40

LIST OF REFERENCES

[1] A. Assad, "Models for Rail Transportation," Transportation Research, 14A, 205-220 (1980).

[2] M. Bellmore and S. Hong, "Transformation of Multisalemen Problem to the Standard Traveling Salesman Problem," Journal of the Association for Computing Machinery, 21, 500-504 (1974).

[3] E. Beltrami and L. Bodin, "Network and Vehicle Routing for Municipal Waste Collection," Networks, 4, 65-94 (1974).

[4] B. Bennett and D. Gazis, "School Bus Routing by Computer," Transportation Research, 6, 317-325 (1972).

[5] N. Christofides and S. Korman, "A Combinatorial Survey of Methods for the Set Covering Problem," Management Science, 21, 591-615 (1975).

[6] N. Christofides, A. Mingozzi, and P. Toth, "The Vehicle Routing Problem," Combinatorial Optimization, Wiley, New York (1979).

[7] N. Christofides, A. Mingozzi, and P. Toth, "State-Space Relaxation Procedures for the Computation of Bounds to Routing Problems," Networks, 11, 145-164 (1981).

[8] N. Christofides and J. Beasley, "The Period Routing Problem," Networks, 14, 237-256 (1984).

[9] N. Christofides, "Vehicle Routing," The Traveling Salesman Problem, Wiley, New York (1985).

[10] G. Clarke and J. Wright, "Scheduling of Vehicles From a Central Depot to a Number of Delivery Points," Operations Research, 12, 568-581 (1964).

[11] G. Dantzig and J. Ramser, "The Truck Dispatching Problem," Management Science, 6, 80-91 (1959).

[12] M. Dessouky, U. Palekar, and H. Zaki, "Development of an Inspector Resource Allocation Model," Interim Report No. 1, Report ORL 87-007 (November 1987).

[13] M. Fisher and R. Jaikumar, "A Generalized Assignment Heuristic for Vehicle Routing," Networks, 11, 109-124 (1981).

[14] R. Garfinkel, "Motivation and Modelling," The Traveling Salesman Problem, Wiley, New York (1985).

[15] W. Garvin, H. Crandall, J. John, and R. Spellman, "Applications of Vehicle Routing in the Oil Industry," Management Science, 3, 407-430 (1957).

[16] B. Gillett and L. Miller, "A Heuristic Algorithm for the Vehicle Dispatch Problem," Operations Research, 22, 340-348 (1974).

[17] B. Golden, T. Magnanti, and H. Nguyen, "Implementing Vehicle Routing Algorithms," Networks, 7, 113-148 (1977).

[18] B.Golden, L. Bodin, T. Doyle, W. Stewart, "Approximate Traveling Salesman Algorithms," Operations Research, 28, 694-711 (1980).

[19] S. Lin, "Computer Solutions to the Traveling Salesman Problem," Bell System Technical Journal, 44, 2245-2269 (1965).

[20] S. Lin and B. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," Operations Research, 21, 498-516 (1973).

[21] T. Magnanti, "Combinatorial Optimization of Vehicle Fleet Planning: Perspectives and Prospects," Networks, 11, 179-213 (1981).

[22] C. Miller, A. Tucker, and R. Zemlin, "Integer Programming Formulation of Traveling Salesman Problems," Journal of the Association for Computing Machinery, 7, 326-329 (1960).

[23] D. Phillips and A. Garcia-Diaz, Fundamentals of Network Analysis, Prentice-Hall, Englewood Cliffs, NJ (1981).

[24] R. Russell, "An Effective Heuristic for the m-Tour Traveling Salesman Problem With Some Side Conditions," Operations Research, 25, 517-524 (1977).

[25] R. Russell and W. Igo, "An Assignment Routing Problem," Networks, 9, 1-17 (1979).

[26] H. Stern and M. Dror, "Routing Electric Meter Readers," Computers and Operations Research, 6, 209-223 (1979).

IMPLEMENTATION AND EVALUATION OF A RESOURCE ALLOCATION ALGORITHM TO DETERMINE THE MINIMUM NUMBER OF INSPECTORS

BY

JOHN T. CLATANOFF

B.S., University of Wisconsin, 1976

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1988

Urbana, Illinois

LIST OF REFERENCES

[1] A. Assad, "Models for Rail Transportation," Transportation Research, 14A, 205-220 (1980).

[2] M. Bellmore and S. Hong, "Transformation of Multisalemen Problem to the Standard Traveling Salesman Problem," Journal of the Association for Computing Machinery, 21, 500-504 (1974).

[3] E. Beltrami and L. Bodin, "Network and Vehicle Routing for Municipal Waste Collection," Networks, 4, 65-94 (1974).

[4] B. Bennett and D. Gazis, "School Bus Routing by Computer," Transportation Research, 6, 317-325 (1972).

[5] N. Christofides and S. Korman, "A Combinatorial Survey of Methods for the Set Covering Problem," Management Science, 21, 591-615 (1975).

[6] N. Christofides, A. Mingozzi, and P. Toth, "The Vehicle Routing Problem," Combinatorial Optimization, Wiley, New York (1979).

[7] N. Christofides, A. Mingozzi, and P. Toth, "State-Space Relaxation Procedures for the Computation of Bounds to Routing Problems," Networks, 11, 145-164 (1981).

[8] N. Christofides and J. Beasley, "The Period Routing Problem," Networks, 14, 237-256 (1984).

[9] N. Christofides, "Vehicle Routing," The Traveling Salesman Problem, Wiley, New York (1985).

[10] G. Clarke and J. Wright, "Scheduling of Vehicles From a Central Depot to a Number of Delivery Points," Operations Research, 12, 568-581 (1964).

[11] G. Dantzig and J. Ramser, "The Truck Dispatching Problem," Management Science, 6, 80-91 (1959).

[12] M. Dessouky, U. Palekar, and H. Zaki, "Development of an Inspector Resource Allocation Model," Interim Report No. 1, Report ORL 87-007 (November 1987).

[13] M. Fisher and R. Jaikumar, "A Generalized Assignment Heuristic for Vehicle Routing," Networks, 11, 109-124 (1981).

[14] R. Garfinkel, "Motivation and Modelling," The Traveling Salesman Problem, Wiley, New York (1985).

[15] W. Garvin, H. Crandall, J. John, and R. Spellman, "Applications of Vehicle Routing in the Oil Industry," Management Science, 3, 407-430 (1957).

[16] B. Gillett and L. Miller, "A Heuristic Algorithm for the Vehicle Dispatch Problem," Operations Research, 22, 340-348 (1974).

[17] B. Golden, T. Magnanti, and H. Nguyen, "Implementing Vehicle Routing Algorithms," Networks, 7, 113-148 (1977).

[18] B.Golden, L. Bodin, T. Doyle, W. Stewart, "Approximate Traveling Salesman Algorithms," Operations Research, 28, 694-711 (1980).

[19] S. Lin, "Computer Solutions to the Traveling Salesman Problem," Bell System Technical Journal, 44, 2245-2269 (1965).

[20] S. Lin and B. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," Operations Research, 21, 498-516 (1973).

[21] T. Magnanti, "Combinatorial Optimization of Vehicle Fleet Planning: Perspectives and Prospects," Networks, 11, 179-213 (1981).

[22] C. Miller, A. Tucker, and R. Zemlin, "Integer Programming Formulation of Traveling Salesman Problems," Journal of the Association for Computing Machinery, 7, 326-329 (1960).

[23] D. Phillips and A. Garcia-Diaz, Fundamentals of Network Analysis, Prentice-Hall, Englewood Cliffs, NJ (1981).

[24] R. Russell, "An Effective Heuristic for the m-Tour Traveling Salesman Problem With Some Side Conditions," Operations Research, 25, 517-524 (1977).

[25] R. Russell and W. Igo, "An Assignment Routing Problem," Networks, 9, 1-17 (1979).

[26] H. Stern and M. Dror, "Routing Electric Meter Readers," Computers and Operations Research, 6, 209-223 (1979).